AFWAL-TR-82-3012

RESEARCH ON A FINITE ELEMENT NUMERICAL ALGORITHM FOR THE THREE-DIMENSIONAL NAVIER-STOKES EQUATIONS

A. J. Baker

Department of Engineering Science and Mechanics
UNIVERSITY OF TENNESSEE
Knoxville, Tennessee

APRIL 1982

Final Report for Period October 1978-September 1981

Flight Dynamics Laboratory
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Charles E. Jobe

CHARLES E. JOBE
Technical Manager
Aerodynamics Methods Group

Lowell C. Keel

LOWELL C. KEEL, Lt Col, USAF
Chief, Aerodynamics & Airframe Branch

FOR THE COMMANDER

John R. Chevalier

JOHN R. CHEVALIER
Colonel, USAF
Chief, Aeromechanics Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIMM *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFWAL - TR - 82 - 3012
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): RESEARCH ON A FINITE ELEMENT NUMERICAL ALGORITHM FOR THE THREE-DIMENSIONAL NAVIER-STOKES EQUATIONS

5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report 1 Oct. 1978 - 30 Sept. 1981

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): A. J. Baker

8. CONTRACT OR GRANT NUMBER(s): AFOSR-79-0005

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Engineering Science and Mechanics, University of Tennessee, Knoxville, TN 37996

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2307N428

11. CONTROLLING OFFICE NAME AND ADDRESS: Flight Dynamics Laboratory (AFWAL/FIMM), AF Wright Aeronautical Laboratory (AFSC), Wright-Patterson AFB, Ohio 45433

12. REPORT DATE: April 1982

13. NUMBER OF PAGES: 112

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

18. SUPPLEMENTARY NOTES:

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):

Navier-Stokes Equations
Finite Element Method
Implicit Solution Algorithm
Generalized Coordinates

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

The objective of this research project was to derive and evaluate accurate and efficient numerical solution algorithms for the three-dimensional Navier-Stokes equations. As a consequence of this objective, a generalized coordinates, implicit finite element numerical algorithm has been established for the problem class. The theoretical basis utilizes a Galerkin-Weighted Residuals statement, rendering the semi-discrete approximation error orthogonal to the finite element subspace, augmented

20. Cont'd

with a penalty constraint forcing orthogonality of the gradient of this error as well. As a consequence, the algorithm possesses highly phase selective dissipation mechanisms permitting accurate resolution of solutions exhibiting a high degree of non-smoothness. A Fourier stability analysis yields an estimate of the dissipation parameter set, which is then refined to enhance the accuracy of a shocked flow prediction.

Multiple factors affecting solution accuracy, convergence and efficiency have been examined. The generalized coordinates framework directly facilitates matching of arbitrary surface descriptions of the solution domain for complete geometric versatility. This formulation as well permits establishment of the tensor matrix product approximation to the Newton iteration algorithm Jacobian, reducing by orders of magnitude the memory and CPU requirements for a multi-dimensional problem definition. The algebraic and constitutive equations, defining pressure, stress tensor and heat flux vector, are handled in an identical manner, yielding an overall consistency to the algorithm. Numerous numerical results are discussed, attesting to the accuracy, convergence, efficiency and versatility of the developed finite element algorithm.

FOREWORD

This report summarizes the results of a three-year study, completed in the Department of Engineering Science and Mechanics, University of Tennessee, on the topic of research on numerical solution algorithms for the three-dimensional Navier-Stokes equations. Sponsorship was provided by the United States Air Force under USAF Grant Number AFOSR-79-0005, Project 2307, Task N4, Work Unit 28.

The principal investigator for this project was Dr. A. J. Baker, Professor of Engineering Science and Mechanics. The contract technical monitor was Dr. Charles E. Jobe, Flight Dynamics Laboratory.

The author wishes to acknowledge the significant contributions of Dr. M. O. Soliman, Assistant Professor of Engineering Science and Mechanics, to this project. Several concepts were refined as a consequence of interaction with valued colleagues, in particular Dr. D. W. Pepper and Mr. J. A. Orzechowski. Extensive support has been provided by the University of Tennessee Computing Center which is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF SYMBOLS

a	boundary condition coefficient; sound speed
A	cross-sectional area; one-dimensional hypermatrix prefix
B	two-dimensional hypermatrix prefix
C	coefficient; matrix; Courant Number; three-dimensional matrix prefix
d	mesh parameter
e	specific total energy
E	energy norm
f	function of known argument
F	finite element matrix; discrete solution equation system
g	total energy
G	finite element matrix
H	Hilbert space
i	$\sqrt{-1}$; index
j	index
J	Jacobian matrix
k	turbulence kinetic energy; cardinal basis degree
ℓ	summation index; differential operator; elemental measure
L	differential operator
m	momentum
M	number of finite elements spanning R^n
n	unit normal vector; dimension of space
N	finite element cardinal basis; discrete index
p	pressure; iteration index
P	intermediate solution matrix
q	heat flux vector; generalized dependent variable

Q	generalized discrete dependent variable
R^n	spatial domain of differential operator
Re	Reynolds Number
S_e	finite element assembly operator
t	time
u_i	velocity vector
$\overline{u_i' u_j'}$	Reynolds kinematic stress tensor
u	convection velocity
x_i	Cartesian coordinate system
α	dependent variable reference
β_i	Lagrange multiplier set
γ	ratio of specific heats; parameter
Γ	phase parameter
∂	partial derivative operator
∂R	boundary of solution domain R^n
δ	Kronecker delta; parameter
δQ	iteration variable
Δ	mesh measure; increment
ε	specific internal energy; isotropic dissipation function; parameter
η_i	curvilinear coordinate system
κ	heat conductivity coefficient
λ	wavelength
μ	dynamic viscosity; dissipation level
ν	kinematic viscosity; dissipation parameter
ρ	density
σ	stress tensor; phase variable

Σ	summation operator
ω	elemental measure; wave number
Ω	solution domain

Superscripts:

h	solution semi-discrete approximation
o	initial condition reference
p	iteration index
T	matrix transpose
1,2	dissipation parameter reference
$'$	ordinary derivative

Subscripts:

e	element reference
i,j,k,ℓ	tensor indices
j	time step index
k	degree of polynomial
o	reference state
α	dependent variable index

Notation:

{ }	column matrix
[ ]	square matrix
$\bigcup$	union
$\bigcap$	intersection
ε	belongs to
$\otimes$	tensor product
$\|\cdot\|$	norm

SECTION I

INTRODUCTION

The past two decades have witnessed a rapidly expanding interest in characterization of high speed flows, principally in aerodynamics. This has fostered formulation and use of numerical algorithm constructions for the governing partial differential equation system. Coupled with recent advances in computer hardware/firmware capabilities, it appears that the age of realistic computational fluid dynamics (CFD) analysis may be nearer at hand. In concert with evolution of the three-color laser velocimeter, both experimental and theoretical fluid dynamics analysis should soon enjoy a greatly expanded resolution capability.

The true "workhorse" of the CFD aerodynamics community over the past decade has been the "MacCormack algorithm" and variations thereof. For example, fully half of the CFD results presented at the recent NASA Lewis Research Center Workshop [1] were generated using this algorithm, originally published in 1968 [2]. This is due principally to the ultimate simplicity of this explicit, predictor-corrector algorithm, and its proven track record for prediction of compressible, supersonic inviscid flowfields. In the split-operator construction, the programming requirements are elementary, and the resultant code runs quite economically. The basic theoretical formulation enjoys continuing refinement, including application to the viscous Navier-Stokes equations, cf. [3,4].

The numerical simulation of viscous or turbulent aerodynamic flows places considerable additional demands on a CFD algorithm. The added discretization refinement, required to resolve wall layers, "stiffens" the resulting discretized equation system. For an elementary explicit algorithm construction, the integration absolute stability interval is bounded by a multiple of the largest eigenvalue of the matrix, yielding the requirement to time march using a very small step size. For this reason at least, and on its own merit, CFD research attention has turned in the last five years to development of implicit Navier-Stokes algorithms. Simply stated, one trades the small time-step restriction for the requirement to solve matrix equation systems. As a result, principal attention has focused on matrix

factorization procedures that reduce large sparse matrices to block-banded forms to permit running solutions on present computer systems.

The approximate-factorization, implicit finite difference (AFFD) Navier-Stokes algorithms of Beam and Warming [5], and Briley and McDonald [6], exemplify the concept. The numerics are typically second-order accurate in space, first order in (pseudo-) time, and may employ both "implicit" and/or "explicit" aritificial diffusion to control perceived instabilities. In the favored non-iterative "delta" formulation, the matrix solution procedure constitutes acceptance of the first iterate of the Newton algorithm, using an approximation to the true matrix system Jacobian, constructed upon addition of the identity matrix.

An integral ingredient of the AFFD algorithm construction is definition and use of the "generalized coordinates" description. Basically, the divergence operator in the Navier-Stokes equation set is transformed into scalar components parallel to principal coordinates of a coordinate transformation "regularizing" the boundary ∂R of the solution domain R^n to the unit square (or cube, n=3). Thompson and coworkers [7] pioneered the transformation concept, the subject of which has become rather popular, cf. [8]. Steger and Pulliam [9] first reported results using the generalized coordinates AFFD algorithm concept for a two-dimensional cascade flow; topical results were recently summarized by Steger [10].

In this same time period, serious attention has become focused on examination of finite element concepts applied to CFD. In its classical application, the "finite element method" is the engineer's utilization of the mathematician's formulation of the variational boundary value problem. Specifically, given a functional describing the state of a system, extremize it with respect to all eligible states, to determine the constraint set of the extremum. The numerical solution of this matrix equation system yields the engineering parameters of interest. Embedded within is the variational calculus, and the formality is applicable to a wide range of problem classes in mechanics and physics, cf. [11, 12].

In its more elementary interpretation, the finite element (FE) method returns calculus and vector field theory to the construction of discrete simulation algorithms for any branch of mechanics. Of necessity, using Tay-

lor series expansions, one must always be able to verify the equivalent (finite difference) order-of-accuracy for elementary derivatives within the governing equation system. Specifically, linear (quadratic) finite elements yield, respectively, second- (fourth-) order accurate finite difference representations for linear, first and second order spatial derivatives. For other than these linear spatial derivatives, the finite element construction usually produces expressions that are not so familiar. However, upon dissection, they can always be related to an appropriate Taylor series expansion , hence be expressed as a difference recursion relation. The important issue is that the theory produces the discrete analog expressions, completely independent of the *a posteriori* ability to construct an equivalent difference representation.

This time frame has also witnessed emergence of the "finite volume" (FV) algorithm, cf. Shang [13]. A FV algorithm is constructed using direct integration of the conservative-form, governing equation system, replacing derivatives by divided (finite) differences. Fluxes are expressed as integrals over the control surfaces of a FV, and scalar fields are typically cell-centered. In an explicit formulation, the FV procedure yields the geometric versatility of a FE discretization, without incurring the coordinate transformation-induced singularities sometimes associated with the generalized coordinates finite difference formulation. An implicit FV algorithm would require handling of large, sparse matrix Jacobians, unless a procedure were invoked to yield the block-banded structure.

The focus of this research project has been to address some of the basic issues regarding theoretical and practical aspects of CFD algorithm constructions, in particular accuracy, convergence and efficiency. The previous two Interim reports [14,15] document certain of these results, and this report summarizes the principal results of the three-year project. There are four basic requirements that must be met in the construction of *any* algorithm for CFD, see Table 1. Denoting the representative partial differential equation of the Navier-Stokes set as $L(q(\vec{x},t))$, the first and most obvious requirement is that an approximation must be specified. The options available include (at least) finite element, finite difference and/or finite volume concepts. Since the approximation cannot be the exact solution, a statement is required regarding what constraints (if any) are placed on the approximation error.

Table 1

Basic Requirements For A CFD Algorithm Construction

Category	Options
1. Approximation	Finite Element (FE) Finite Difference (FD) Finite Volume (FV)
2. Approximation Error	Orthogonality (FE) Taylor Series (FD, FV)
3. Dispersion Error	Multi-pole Expansion (FE) Artificial Diffusion (FD, FV)
4. Matrix Solution Jacobian	Tensor Matrix Products (FE) Approximate Factorization (FD)

The FD and FV constructions differ from FE concepts in this specific regard, in that categories 1 and 2 are grouped together in replacing $L(q(\vec{x},t))$ with difference quotients yielding $L(Q(\Delta\vec{x}_j,\Delta t)) \equiv 0$. In the FE construction of category 1, the approximation is made to the solution rather than $L(q(\vec{x},t))$. A vast reservoir exists, from which one can select particular members that endow the solution approximation $q^h(\vec{x},t)$ with specific properties, eg., completeness. Then, $L(q^h(\vec{x},t)) \neq 0$ is the error in the governing differential equation. The category 2 specification simply requires this error to be orthogonal (perpendicular) to the function space from which q^h was formed. Calculus and vector field operations permit an exact evaluation of this constraint, yielding the matrix equivalent statement $L(\{Q(t)\}) \equiv \{0\}$.

For a viscous-dominated (small Reynolds number) flow, the category 3 issue of dispersion error need not be addressed. However, most aerodynamic flows are characterized by a large Reynolds number, yielding the governing-Navier-Stokes equations elliptic only in small regions of R^n directly adjacent to aerodynamic surfaces. Elsewhere, the equation set is dominantly

hyperbolic and non-linear. The principal error mechanism operating in discrete approximations to hyperbolic equations is dispersion error, the tendency of the Fourier components constituting information packets to possess distinct phase velocities differing from the group velocity. The Navier-Stokes non-linearity aggravates this character, yielding a divergence that can eventually "blow-up" the solution. The algorithm requirement is to control (diffuse) this error mechanism. The standard FD(FV) practice is to explicitly add a diffusion term, multiplied by a coefficient of "artificial viscosity," to the governing differential equation $L(q(\vec{x},t))$. The concept, first formalized by von Neumann and Richtmyer [16], is broadly used. The FE basis for this error control is recognition that the approximation error $L(q^h(\vec{x},t)$ will be highly non-smooth in regions where dispersion error is excessive. Interpreting $L(q^h(\vec{x},t))$ as a (hyper-) surface, regions of large error are characterized by large changes in surface amplitude and direction of the normal. Noting that $\nabla L(q^h(\vec{x},t))$ characterizes both the magnitude and direction of the non-smoothness, the FE concept requires this error also be orthogonal to the approximation sub-space, subject to a constraint set $\vec{\beta}$. This of course yields an "artificial viscosity" term, as well as some additional terms, and it is interesting to compare performance of the FD and FE formalisms.

The completion of categories 1-3 yields a matrix equation system {F}, that is non-linear unless explicit time-integration has been specified. In all other instances, including a direct steady-state algorithm, and as a function of the dimension n of R^n and the number of dependent variables defined in L(q), the Jacobian of {F} is a *large* albeit sparse matrix. The direct matrix solution using this Jacobian is usually ill-advised, based strictly on the operation count, core storage and CPU requirements. One FD approach is to construct an approximate factorization, as discussed earlier. One FE approach, presented and discussed herein, is to construct the tensor (outer) matrix product of the Jacobian. Of course, since category 4 is strictly a problem in linear algebra, there exists a wide range of alternative procedures including SOR, SLOR, and ADI to name a few. The selection of any of these can be a (the) prime determining factor in the *cost* of obtaining the solution. The solution cost is typically much less affected by choices made in categories 1-3.

Before proceeding into the detailed technical analysis, an amplification on category 3 is appropriate. The concept of "artificial viscosity" has been rediscovered innumerable times, under such euphemisms as upwinding, upstream differencing, donor cell, streamline upwinding, Petrov-Galerkin, non-equal weighting (and multi-pole expansion). In every instance, terms are added to *smooth* the solution. There has also been a tendency to prescribe a dose of artificial viscosity to smooth error introduced by inadequate grid resolution, distorted grids, and inconsistent formulations, or to "remedy" the overall poor phase accuracy of the specific algorithm. Strictly speaking, the use of fictitious diffusion to control other than non-linear induced instability constitutes an additional error source not covered in Table 1. The "suppression of the wiggles", cf. [17], really constitutes elimination of information about the approximate solution, specifically its inadequacy. This facet of CFD certainly deserves close attention, in the context of a missing category in Table 1.

SECTION II

PROBLEM STATEMENT

The partial differential equation set governing three-dimensional aerodynamic flows is the familiar and very non-linear Navier-Stokes system. In non-dimensional conservation form, using Cartesian tensor summation notation, the equation set governing flow of a compressible, viscous, heat-conducting fluid is

$$L(\rho) = \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}[u_j \rho] = 0 \tag{1}$$

$$L(\rho u_i) = \frac{\partial (\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}\left[u_j\, \rho u_i + p\delta_{ij} - \sigma_{ij}\right] = 0 \tag{2}$$

$$L(\rho e) = \frac{\partial (\rho e)}{\partial t} + \frac{\partial}{\partial x_j}[u_j \rho e + u_j p - \sigma_{ij} u_i - q_j] = 0 \tag{3}$$

In equations 1-3, ρ is density, $\rho u_i \equiv m_i$ is the momentum vector, p is pressure, and e is mass specific total energy. The Stokes viscous stress tensor σ_{ij} and heat flux vector q_j, in terms of specific internal energy ε, are

$$\sigma_{ij} = \frac{\mu}{Re}\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right] - \frac{\mu}{3Re}\frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{4}$$

$$q_j = -\kappa \frac{\partial \varepsilon}{\partial x_j} \tag{5}$$

$$\varepsilon = e - \frac{1}{2} u_i u_i \tag{6}$$

Assuming a polytropic gas, $p = (\gamma-1)\rho\varepsilon$, and the equation of state is

$$L(p) = p - (\gamma-1)[\rho e - \tfrac{1}{2}\rho u_j u_j] = 0 \tag{7}$$

Finally, μ is the absolute viscosity, κ is the coefficient of heat conductivity, δ_{ij} is the Kronecker delta, and Re is the reference Reynolds number.

The Euler equations are contained within this description by the specification that equations 4 and 5 vanish identically. The form of equations 1-3 is also representative of the mass-weighted formulation of the time-averaged Navier-Stokes equations for a turbulent flow [18]. The dependent variables are interpreted as descriptors of the mean flow, and σ_{ij} is generalized to include nonvanishing correlations of sub-grid scale phenomena. In this instance, the total stress tensor becomes of the form,

$$\sigma_{ij} \equiv \bar{\sigma}_{ij} - \overline{\rho u_i' u_j'} \tag{8}$$

where $-\overline{\rho u_i' u_j'}$ is the dynamic Reynolds stress tensor, and $\bar{\sigma}_{ij}$ denotes the time averaged form for equation 4.

Procedures for closing equation 8 include solution of the Reynolds stress transport equations, cf. [18,19], tensor field constitutive theory [20], as well as elementary mixing length theory [18]. For the first closure, σ_{ij} will not be an explicit function of m_i; for the latter two, it becomes a strongly non-linear function of m_i. The field theory approach encompasses the latter, in the form

$$\overline{u_i' u_j'} = C_1 k \delta_{ij} - C_\nu^1 \frac{k^2}{\varepsilon} \left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right] - C_\nu^2 \frac{k^3}{\varepsilon} \left[\frac{\partial \bar{u}_i}{\partial x_\ell} + \frac{\partial \bar{u}_\ell}{\partial x_i} \right] \left[\frac{\partial \bar{u}_\ell}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_\ell} \right] + \ldots \tag{9}$$

The various correlation coefficients C_ν^α are defined [20], and the turbulence kinetic energy and isotropic dissipation function are, respectively

$$k \equiv \tfrac{1}{2} \overline{u_i' u_i'} \tag{10}$$

$$\varepsilon \equiv \frac{2\nu}{3} \overline{\left[\frac{\partial u_i'}{\partial x_j} \frac{\partial u_i'}{\partial x_k} \right]} \delta_{jk} \tag{11}$$

The variables k and ε are solutions to the corresponding transport differential equations [19],

$$L(k) = \frac{\partial(\bar{\rho}k)}{\partial t} + \frac{\partial}{\partial x_j}\left[\bar{\rho}\tilde{u}_j k + \left(C_k\,\overline{\rho u_i'' u_j''}\,\frac{k}{\varepsilon} - \bar{\mu}\delta_{ij}\right)\frac{\partial k}{\partial x_i}\right]$$

$$+ \overline{\rho u_i'' u_j''}\,\frac{\partial \bar{u}_i}{\partial x_j} + \bar{\rho}\varepsilon = 0 \qquad (12)$$

$$L(\varepsilon) = \frac{\partial(\bar{\rho}\varepsilon)}{\partial t} + \frac{\partial}{\partial x_j}\left[\bar{\rho}\tilde{u}_j\varepsilon + \left(C_\varepsilon\,\overline{\rho u_i'' u_j''}\,\frac{k}{\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_i}\right]$$

$$+ C_\varepsilon^1\,\overline{\rho u_i'' u_j''}\,\frac{\varepsilon}{k}\,\frac{\partial \bar{u}_i}{\partial x_j} + C_\varepsilon^2\bar{\rho}\frac{\varepsilon^2}{k} = 0 \qquad (13)$$

The correlation coefficients C_k and C_ε^α have been determined from analysis and experiment [19].

The boundary conditions for the equation set 1-13 are a general mixture of Dirichlet and Neumann specifications. On an inflow boundary segment, ρ, ρu_i, ρe, k and ε are typically specified and p is determined from equation 6. For an inviscid streamline (Euler equations on a farfield boundary), the normal derivative of the scalar variables vanish, and ρu_i is constrained to be parallel. For a no-slip aerodynamic surface, ρu_i, k and ε vanish identically, and the normal derivative of Θ is constrained (adiabatic or cooled wall). At an outflow boundary, typically p is specified and the normal derivative of the remaining dependent variables is constrained.

SECTION III

FINITE ELEMENT SOLUTION ALGORITHM

1. Theoretical Statement

Equations 1-13 constitute a fairly general definition of the three-dimensional Navier-Stokes equations. In the limit Re => ∞, equation 4, they reduce to the Euler equations governing an inviscid rotational flow. Specifying $m_i \equiv \rho u_i$, $g \equiv \rho e$, define the vector-valued dependent variable set as

$$q_\alpha(x_i,t) \equiv \{q_\alpha\} \equiv \{\rho, m_i, g, p, \sigma_{ij}, q_j, \bar{\rho}k, \bar{\rho}\varepsilon\} \tag{14}$$

Equations 1-5, 12, 13 constitute an initial-value, elliptic boundary value description of the general form

$$L(q_\alpha) = \frac{\partial q_\alpha}{\partial t} + \frac{\partial}{\partial x_j}\left[u_j q_\alpha + f_{\alpha j}\right] + s_\alpha = 0 \tag{15}$$

In equation 15, $f_{\alpha j}(q_\beta)$ and $s_\alpha(q_\beta)$ are specific non-linear functions of their argument, as determined by inspection. The remaining algebraic and partial differential equations 7-9 are of the form

$$L(q_\alpha) = q_\alpha + f_\alpha(q_\beta) = 0 \tag{16}$$

where the f_α are also determined by inspection. Note that equation 16 is simply a special form of equation 15.

The n-dimensional partial differential equation system 15-16 is defined on the Euclidean space R^n, spanned by the $\vec{x}$ coordinate system with scalar components x_i, $1 \leq i \leq n$. The solution domain Ω is defined as the product of R^n and t, for all elements of $\vec{x}$ belonging to R^n and all elements of t belonging to the open interval measured from t_o, i.e.,

$$\Omega \equiv R^n \times t = \{(\vec{x},t): \vec{x}\varepsilon R^n \text{ and } t\varepsilon[t_o,t)\} \tag{17}$$

The boundary $\partial\Omega$ of the solution domain is the product of the boundary ∂R of R^n, spanned by $\vec{x}$, and t, i.e., $\partial\Omega = \partial R \times t$. Thereupon, the generalized form for the differential boundary constraint is

$$\ell(q_\alpha) = a_1^\alpha q_\alpha + a_2^\alpha \frac{\partial}{\partial x_j} q_\alpha \hat{n}_j + a_3^\alpha = 0 \tag{18}$$

In equation 18, the a_i^α are specified coefficients and $\hat{n}_j$ is the outwards pointing unit normal vector. Finally, an initial distribution for q_α on $\Omega_o = R^n \times t_o$ is required.

$$q_\alpha(\vec{x},t_o) = q_\alpha^o(\vec{x}) \tag{19}$$

The functional requirement of the (any) numerical solution algorithm for equations 14-19 is to construct a suitable "approximation" to the dependent variable set $q_\alpha(\vec{x},t)$, and to render the error in this approximation extremum in some norm, recall Table 1. A finite element algorithm construction is simply the formalized application of this basic requirement, as discussed. The semi-discrete approximation $q_\alpha^h(\vec{x},t)$ to the unknown exact solution $q_\alpha(\vec{x},t)$ is constructed from members of a convenient finite-dimensional subspace of $H_0^1(R)$, the Hilbert space of all functions possessing square integrable first derivatives and satisfying the boundary conditions, equation 18. While extremely flexible in theory, the typical practice is to employ elementary polynomials, truncated at degree k, and defined on disjoint interior subdomains R_e^n, the union of which forms the discretization of $R^n \equiv UR_e^n$. Hence,

$$q_\alpha(\vec{x},t) \approx q_\alpha^h(\vec{x},t) = \sum_{e=1}^{M} q_\alpha^e(\vec{x},t) \tag{20}$$

and

$$q_\alpha^e(\vec{x},t)\} \equiv \{N_k(\vec{x})\}^T \{QI(t)\}_e \tag{21}$$

In equation 21, the semi-discrete free index "I" denotes q_α^h evaluated at the nodal coordinates of the discretization UR_e^n at anytime t. The sub- or

superscript e indicates pertaining to the $e^{\underline{th}}$ finite element domain, $\Omega_e = R^n_e xt$. The elements of the row matrix $\{N_k(\vec{x})\}^T$ are polynomials written on x_j, $1 \leq j \leq n$, complete to degree k and constructed to form a cardinal basis [21].

The fundamental requirement of any numerical algorithm is to render the (semi-discrete) approximation error minimum in some norm. For smooth solutions, this is accomplished within a finite element context by requiring the error in equations 15, 16 and 18, ie., $L(q^h_\alpha)$ and $\ell(q^h_\alpha)$, to be orthogonal to the function space used to construct q^h_α. This basic requirement may be augmented, in the sense of a multi-pole expansion, to enforce further constraints on the algorithm construction. For example, for high Mach number flows wherein non-smooth solutions may exist, the dissipation mechanism for control of non-linearly induced instabilities is generated by also requiring the error (vector) gradient $\nabla L(q^h_\alpha)$ to be orthogonal to $\{N_k\}$. As a second example, for certain steady, subsonic flows, the solution to the momentum equation 2 can be constrained [20] by the requirement that the velocity field be divergence free, equation 1. Each of these constitutes a term in a multi-pole expansion of the basic error orthogonalization. Defining the (Lagrange) multiplier set $\vec{\beta}_i$ permits adjoining these linearly independent constraints, yielding the finite element numerical solution algorithm statement for the compressible Navier-Stokes equations as

$$\int_{R^n}\{N_k\}L(q^h_\alpha)d\vec{x} + \beta_1\int_{\partial R}\{N_k\}\ell(q^h_\alpha)d\vec{x} + \vec{\beta}_2\cdot\int_{R^n}\{N_k\}\nabla L(q^h_\alpha)d\vec{x} \equiv \{0\} \qquad (22)$$

Upon definition of k in equation 21, equation 22 is a mixed system of ordinary differential and algebraic equations, written on t, of the form

$$[C]\{QI\}' + [U]\{QI\} + [FIJ]\{QJ\} + \{SI\} \equiv \{0\} \qquad (23)$$

A one-to-one correspondence of terms in equations 23 and 15-16 is inferred, with the matrices in equation 23 augmented to contain the various additional terms introduced through $\vec{\beta}_i \neq 0$ in equation 22. (Detailed expansions are presented in a later section.) Since the [C] matrix is non-diagonal, the

form of equation 23 suggests use of an implicit integration algorithm. The efficient Θ-implicit finite difference algorithm, where $\Theta = \frac{1}{2}$ yields the trapezoidal rule, is

$$\{FI\} = \{QI\}_{j+1} - \{QI\}_j - \Delta t\left[\Theta\{QI\}'_{j+1} + (1-\Theta)\{QI\}'_j\right] \equiv 0 \quad (24)$$

Equation 23 provides the definition of the derivatives $\{QI\}'$. Upon substitution, and proceeding through the algebra [14], equation 24 becomes a nonlinear algebraic equation system written on $\{QI\}_{j+1}$. The Newton iteration algorithm solution statement is

$$[J(FI)]^p_{j+1}\{\delta QI\}^{p+1}_{j+1} = -\{FI\}^p_{j+1} \quad (25)$$

The dependent variable in equation 25 is the iteration vector $\{\delta QI\}$, related to the solution in the conventional manner.

$$\{QI\}^{p+1}_{j+1} \equiv \{QI\}^p_{j+1} + \{\delta QI\}^{p+1}_{j+1} \quad (26)$$

The Jacobian is defined as,

$$[J(FI)] \equiv \frac{\partial\{FI\}}{\partial\{QJ\}} \quad (27)$$

The detailed construction is direct, upon expansion of equations 22-24 using a hypermatrix formulation, to be discussed.

2. Generalized Coordinates

A principal requirement in computational aerodynamics is to accurately interpolate domain boundary geometries $\partial\Omega$ that are non-regular and perhaps non-smooth. The term "generalized coordinates" has gained acceptance to describe the algorithm statement appropriate for use with a regularizing, boundary-fitted coordinate transformation. Many procedures are available to construct such transformations [8], including numerical solution of elliptic and hyperbolic partial differential equations and algebraic methods.

The output of any of these procedures constitutes definition of coordinate triples (pairs) on R^3 (R^2), that define the intersections of ∂R_e on Ω called nodes. The computational requirement is to construct a local coordinate transformation, using these data, that maps $x^i \epsilon R_e^n$ to a regularized domain spanned by an orthonormal coordinate system η_j. Figure 1 illustrates the concept, where (•) and (x) depict the nodal coordinates in both spaces.

The required coordinate transformation is.

$$x_i = x_i(\eta_j) \tag{28}$$

The explicit form for equation 28 is constructed using elementary interpolation concepts and the cardinal basis $\{N_k\}$. Since equation 21 defines any approximation, the form for equation 28 is

$$x_i = \{N_k(\vec{\eta})\}^T \{XI\}_e, \qquad x_i \epsilon R_e^n \tag{29}$$

a) Two-Dimensional Domain

b) Three-Dimensional Domain

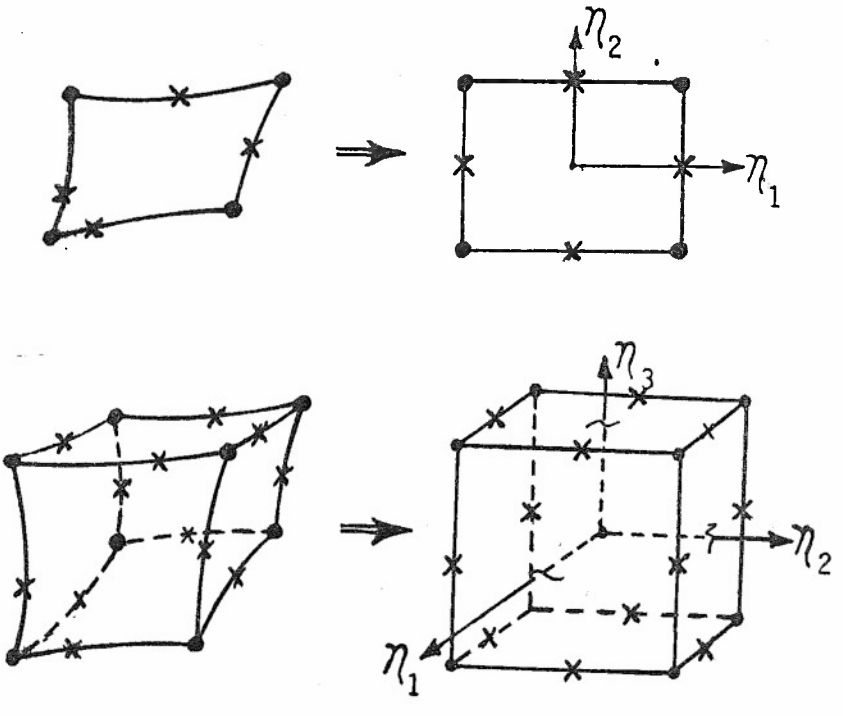


Figure 1: Biquadratic Cardinal Basis Coordinate Transformation

The elements of the column matrix $\{XI\}_e$ are the coordinates of the nodes of R_e^n, $1 \leq (i,I) \leq n$, hence UR_e^n, the global discretization.

Assuming the discretization of R^n adequately represented by plane-faced hexahedra (or straight-sided quadrilaterals, for R^2), the k=1 basis $\{N_1(\vec{\eta})\}$ yields the exact representation. For example, on R^2,

$$\{N_1(\vec{\eta})\} = \frac{1}{4} \begin{Bmatrix} (1-\eta_1)(1-\eta_2) \\ (1+\eta_1)(1-\eta_2) \\ (1+\eta_1)(1+\eta_2) \\ (1-\eta_1)(1+\eta_2) \end{Bmatrix} \tag{30}$$

A discretization on R^2 using curved-sided quadrilaterals must contain at least eight nodes per element, and

$$\{N_2(\vec{\eta})\} = \frac{1}{4} \begin{Bmatrix} (1-\eta_1)(1-\eta_2)(-\eta_1-\eta_2-1) \\ (1+\eta_1)(1-\eta_2)(\eta_1-\eta_2-1) \\ (1+\eta_1)(1+\eta_2)(\eta_1+\eta_2-1) \\ (1-\eta_1)(1+\eta_2)(-\eta_1+\eta_2-1) \\ (1-\eta_1)(1-\eta_2) \\ (1+\eta_1)(1-\eta_2) \\ (1+\eta_1)(1+\eta_2) \\ (1-\eta_1)(1+\eta_2) \end{Bmatrix} \tag{31}$$

For discretizations of R^3 employing planar-faced hexahedra, the corresponding cardinal basis is

$$\{N_1(\vec{\eta})\} = \frac{1}{8} \begin{Bmatrix} (1-\eta_1)(1-\eta_2)(1-\eta_3) \\ (1+\eta_1)(1-\eta_2)(1-\eta_3) \\ (1+\eta_1)(1+\eta_2)(1-\eta_3) \\ (1-\eta_1)(1+\eta_2)(1-\eta_3) \\ (1-\eta_1)(1-\eta_2)(1+\eta_3) \\ (1+\eta_1)(1-\eta_2)(1+\eta_3) \\ (1+\eta_1)(1+\eta_2)(1+\eta_3) \\ (1-\eta_1)(1+\eta_2)(1+\eta_3) \end{Bmatrix} \tag{32}$$

Returning to the issue, the generalized-coordinates algorithm requirement is transformation of the divergence operator in equation 15, i.e.,

$$\frac{\partial}{\partial x_i} = \frac{\partial \eta_j}{\partial x_i} \frac{\partial}{\partial \eta_j} \tag{33}$$

The elements of the inverse Jacobian $J^{-1} \equiv [\partial \eta_i / \partial x_j]$ are,

$$\left[\frac{\partial \eta_i}{\partial x_j}\right] \equiv J^{-1} = \frac{1}{\det[J]} \text{ [transformed cofactor of J]} \qquad (34)$$

J is the Jacobian of the forward transformation, equation 28,

$$[J] \equiv \left[\frac{\partial x_i}{\partial \eta_j}\right] \qquad (35)$$

which is directly evaluated in terms of the nodal coordinates of R_e^n using equation 29. The differential element for equation 22 is

$$d\vec{x} = \det[J] d\vec{\eta} \qquad (36)$$

As an example, consider equation 22 written for the momentum equation 2. Using a Green-Gauss form of the divergence theorem, the first constraint term of equation 22 becomes

$$\begin{aligned} \int_{R^n} \{N\} L(\rho u_i^h) d\vec{x} &= \int_{R^n} \{N\} \frac{\partial \rho u_i^h}{\partial t} \det[J] d\vec{\eta} \\ &+ \oint_{\partial R} \{N\} \left[u_j^h \rho u_i^h + p^h \delta_{ij} - \sigma_{ij}^h\right] \cdot \hat{n}_j \det[J] d\vec{\eta} \\ &- \int_{R^n} \frac{\partial \{N\}}{\partial \eta_k} \left(\frac{\partial \eta_k}{\partial x_j}\right) \left[u_j^h \rho u_i^h + p^h \delta_{ij} - \sigma_{ij}^h\right] \det[J] d\vec{\eta} \end{aligned} \qquad (37)$$

Define the contravariant components of the convection velocity as,

$$\tilde{u}_k^h \equiv \det[J] \left(\frac{\partial \eta_k}{\partial x_j}\right) u_j^h \qquad (38)$$

with scalar components parallel to the η_k coordinate system. Using the algebraic transformation equation 29, det[J] cancels yielding

$$\bar{u}_k^h = [\text{Cof. } J]\, u_j^h \tag{39}$$

where [Cof. J] is the transformed co-factor matrix of [J]. Using equation 21 to interpolate the nodal distribution of $\bar{u}_k^h$ on R_e^n, equation 37 becomes

$$\begin{aligned}\int_{R^n}\{N\}L(\rho u_i^h)dx \equiv S_e\Bigg[&\int_{R_e^n}\{DET\}_e^T\{N\}\{N\}\{N\}^T\{RHOUI\}_e\dot{d\vec{\eta}}\\ &-\int_{R_e^n}\{UBARK\}_e^T\{N\}\frac{\partial}{\partial\eta_k}\{N\}\{N\}^T\{RHOUI\}_e d\vec{\eta}\\ &-\int_{R_e^n}\{ETAKI\}_e^T\{N\}\frac{\partial}{\partial\eta_k}\{N\}\{N\}^T\{P\}_e d\vec{\eta}\\ &+\int_{R_e^n}\{ETAKJ\}_e^T\{N\}\frac{\partial}{\partial\eta_k}\{N\}\{N\}^T\{SIGIJ\}_e d\vec{\eta}\Bigg]\\ +\int_{\partial R_e\cap\partial R}\Big(&\{UBARK\}_e^T\{N\}\{N\}\{N\}^T\{RHOUI\}_e+\{DET\}_e^T\{N\}\{N\}\{N\}^T(\{P\}_e\delta_{IK}-\{SIGIK\}_e)\Big)\cdot\hat{n}_k d\vec{\eta}\end{aligned} \tag{40}$$

In equation 40, since the $\{N_k\}$ for q_α^h are defined locally on R_e^n, the limits for the defined integrals are ±1. S_e is the matrix operator projecting computed element contributions to the corresponding global matrices, equation 22. The determinant of J is interpolated on the element R_e^n using $\{N_k\}$ and the nodal values. Similarly, $\partial\eta_k/\partial x_i$ is interpolated as $\{ETAKI\}_e^T\{N_k\}$.

The e-subscripted terms in equation 40 are independent of η_k, hence can be extracted from the integrand leaving only products of the polynomials and derivatives of $\{N_k\}$, cf. equations 30-32. These are directly evaluated using numerical quadrature. Hence, the finite element algorithm statement for the first term in equation 22, as formed for equation 2, is

$$\begin{aligned}\int_{R^n}\{N\}L(\rho u_i^h)dx = S_e\Bigg[&\{DET\}_e^T[M3000]\{RHOUI\}_e\dot{} - \{UBARK\}_e^T[M30K0]\{RHOUI\}_e\\ &-\{ETAKI\}_e^T[M30K0]\{P\}_e+\{ETAKL\}_e^T[M30K0]\{SIGIL\}_e\\ &+\hat{n}_k\Big(\{UBARK\}_e^T[B3000]\{RHOUI\}+\{DET\}_e^T[B3000]\{P\}_e\delta_{IK}\\ &-\{DET\}_e^T[B3000]\{SIGIK\}_e\Big)\Bigg]\end{aligned} \tag{41}$$

In equation 41, the indices K and L obey the tensor summation rule, I is the free index for ρu_i^h, S_e is the assembly operator, and [M30K0] is the element-independent hypermatrix equivalent of $\partial/\partial\eta_k$ integrated over R_e^n. The matrix [B3000] results from interpolation integrals on the element-boundary intersection $\partial R_e \cap \partial R$ with outward normal $\hat{n}_k$. For the coordinate transformation, $\{DET\}_e$ is the nodal distribution of det[J] and $\{ETAKI\}_e$ and $\{ETAKL\}_e$ are corresponding nodal distributions of components of J^{-1} on R_e^n. Within the generalized coordinate framework, therefore, the grid and metric data required for a numerical simulation are the nodal distributions of $J^{-1} = [\partial\eta_k/\partial x_j]$, and $\det[J] = \det[\partial x_i/\partial\eta_k]$. These data are strictly a function of the nodal coordinate distribution of the discretization UR_e^n, see equation 28-29, and may be provided using any coordinate transformation procedure, cf. [8].

3. Tensor Matrix Product Jacobian

For efficiency, recall category 4 of Table 1, it is desirable to construct a suitable approximation to the equation system Jacobian defined in equation 25, see equation 27. The matrix tensor product construction is facilitated by using the tensor product cardinal basis function set $\{N_k(\vec{\eta})\}$ spanning quadrilateral and hexahedra element domains R_e^2 and R_e^3 respectively. The Jacobian matrix [J(FI)], equation 27, is then replaced by the tensor (outer) product construction [22].

$$[J(FI)] \Rightarrow [J_1] \otimes [J_2] \otimes [J_3] \tag{42}$$

Each component $[J_\eta]$ is constructed from its definition, equation 27, assuming interpolation and differentiation are one-dimensional. Using equation 42, the matrix statement equation 25 becomes

$$[J_1] \otimes [J_2] \otimes [J_3] \{\delta QI\}_{j+1}^{p+1} = - \{FI\}_{j+1}^{p} \tag{43}$$

Define

$$[J_2] \otimes [J_3] \{\delta QI\}_{j+1}^{p+1} \equiv \{P1\}_{j+1}^{p+1}$$

$$[J_3]\{\delta QI\}_{j+1}^{p+1} \equiv \{P2\}_{j+1}^{p+1} \tag{44}$$

Then, the operation defined by equation 43 is the sequence.

$$\begin{aligned}
[J_1]\{P1\}_{j+1}^{p+1} &= -\{FI\}_{j+1}^{p} \\
[J_2]\{P2\}_{j+1}^{p+1} &= \{P1\}_{j+1}^{p+1} \\
[J_3]\{\delta QI\}_{j+1}^{p+1} &= \{P2\}_{j+1}^{p+1}
\end{aligned} \tag{45}$$

Obviously, other permutations of the index structure for $[J_\eta]$ could be utilized. The key aspect is replacement of the very large (albeit sparse) Jacobian matrix [J], with α-block-structured matrices $[J_\eta]$. The principal attributes are several orders of magnitude reduction in central memory requirements for the Jacobian, and significantly reduced CPU for the LU decomposition and back substitution solution steps. The principal detraction is potential degradation of the quadratic convergence rate for the Newton iteration, equation 25. This procedure in no way affects the formation of {FI}, equation 24, wherein lies the accuracy features intrinsic to the finite element algorithm statement. Compromises in the evaluation of {FI} will invariably produce inferior results for the Navier-Stokes equations.

The construction of the $[J_\eta]$ is straightforward. For example, for the initial-value term in equation 41, the corresponding term expressing self-coupling in the Jacobian is

$$\frac{\partial\{FI\}}{\partial\{QJ\}} \equiv [JQQ] = S_e\left[\{DET\}_e^T[M3000]\delta_{IJ}\right] \tag{46}$$

where δ_{IJ} is the discrete index Kronecker delta. Referring to equation 40 for the definition, the elemental operation in forming equation 46 is

$$[JQQ]_e \equiv \{DET\}_e^T[M3000] \equiv \int_{R_e^n} \{DET\}_e^T\{N_k(\vec{\eta})\}\{N_k(\vec{\eta})\}\{N_k(\vec{\eta})\}^T d\vec{\eta} \tag{47}$$

Assume for simplicity the most elementary case, i.e., k=1, n=2 and $\vec{x} \equiv \vec{\eta}$, i.e., the identity transformation. Defining M ≡ B for n=2, equation 47 becomes

$$\{DET\}_e^T[B3000] = \Delta_e[B200] \equiv \int_{R_e^2} \{N_1(\vec{x})\}\{N_1(\vec{x})\}^T d\vec{x} \tag{48}$$

Assuming the rectangular element domain R_e^2 described by measures ℓ and ω, the evaluation of equation 48 yields

$$[JQQ]_e = \Delta_e[B200] = \ell\omega[B200] = \frac{\ell\omega}{36}\begin{bmatrix} 4 & 2 & 1 & 2 \\ & 4 & 2 & 1 \\ & & 4 & 2 \\ (sym) & & & 4 \end{bmatrix} \tag{49}$$

The tensor product construction for this matrix involves evaluation of equation 48 constrained to one-dimension. Denoting M ≡ A for n=1,

$$[JQQ_\eta]_e \equiv \Delta_\eta[A200] = \int_{R_e^1} \{N_1(x_1)\}\{N_1(x_1)\}^T dx_1 \tag{50}$$

and

$$[JQQ_1]_e = \frac{\ell}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$[JQQ_2]_e = \frac{\omega}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{51}$$

assuming $\Delta_1 = \ell$ and $\Delta_2 = \omega$. Accounting for entry locations in [JQQ], using the assembly operator S_e, cf. [23], it is readily verified that

$$[JQQ] = S_e\left[[JQQ_1]_e \otimes [JQQ_2]_e\right] \tag{52}$$

By direct extension, then, the tensor matrix product form of equation 46 on R^3 is

$$S_e\left[\{DET\}_e^T[M3000]\right] \Rightarrow S_e\left[\{DET1\}_e^T[A3000] \otimes \{DET2\}_e^T[A3000] \otimes \{DET3\}^T[A3000]\right] \tag{53}$$

In equation 53, the index $1 \leq K \leq 3$ on $\{DETK\}_e$ denotes the scalar component of the multi-dimensional measure parallel to η_k.

The second term in the typical Jacobian, as developed in Section III.4, is the form,

$$S_e\left[\nu^1_{\alpha J}\{ETAKJ\}^T_e[M40K00]\{DET\}_e\right]$$

Here $\nu^1_{\alpha J}$ results from definition of the specific form for $\vec{\beta}_2$, equation 22, and α denotes the specific dependent variable. The indices $1 \leq (K,J) \leq n$ are discrete tensor summation indices with range $1 \leq (K, J) \leq n$. [M40K00] is a hypermatrix of degree two, ie., it possesses elements which are themselves square matrices, which represents the integration over R^n_e of four cardinal basis $\{N_k(\vec{\eta})\}$, one of which is differentiated with scalar components parallel to η_k. The tensor matrix product equivalent is

$$S_e\left[\nu^1_{\alpha J}\{ETAKJ\}^T_e[M40K00]\{DET\}_e\right]$$

$$\Rightarrow S_e\left[\nu^1_{\alpha J}\{ETA1J\}^T_e[A40100]\{DET1\}_e \otimes \{ETA2J\}^T_e\ [A40100]\{DET2\}_e \otimes \right.$$

$$\left. \{ETA3J\}^T_e[A40100]\{DET\}_e\right] \qquad (54)$$

In equation 54, J remains a discrete summation index with range $1 \leq J \leq n$.

4. Algorithm Matrix Statement

As developed in the next Section, $\vec{\beta}_2 \equiv \{DET\}_e\, \nu^\gamma_{\alpha J}$ is the functional form for the multi-pole term in equation 22. With these identifications, the specific matrix forms of the algorithm statement $\{FI\} = \{0\}$, equation 24, and the tensor product Jacobian $[J_\eta]$, equations 27 and 42, can be expressed. In all cases, the global matrix statement is obtained using S_e on the sequence of elemental definitions, e.g.,

$$\{FI\} \equiv S_e\left[\{FI\}_e\right] \qquad (55)$$

The elemental expression $\{FI\}_e$ is always obtained using the element-independent master hypermatrices [M...], contracted with element-dependent matrices. In the following expressions, subscripts "e" are omitted in the expansions for clarity.

Denote the members of the discrete dependent variable set as $\{QI\}^T$ = {R, M(I), G, P, S(I,J), Q(J), RK, RE}, see equation 14. The matrix algorithm statements $\{FI\}_e$, equation 55, for arbitrary basis $\{N_k\}$ and dimension n, are:

$$\{FR\}_e = \left[\{DET\}^T[M3000] + \nu^1_{RJ}\{ETAKJ\}^T[M40K00]\{DET\}\right]\{R\}'_{j+1}$$
$$+ \frac{\Delta t}{2}\left[-\{ETAKI\}^T[M30K0]\{MI\} + \nu^2_{RJ}\{ETAKJ\}^T[M40KL0]\{UBARL\}\{R\}\right]_{j+1,j} \quad (56)$$

$$\{FMI\}_e = \left[\{DET\}^T[M3000] + \nu^1_{IJ}\{ETAKJ\}^T[M40K00]\{DET\}\right]\{MI\}'_{j+1}$$
$$+ \frac{\Delta t}{2}\Big[-\{UBARK\}^T[M30K0]\{MI\} - \{ETAKI\}^T[M30K0]\{P\}$$
$$+ \{ETAKJ\}^T[M30K0]\{SIGIJ\}$$
$$+ \nu^2_{IJ}\{ETAKJ\}^T[M40KL0]\{UBARL\}\{MI\}\Big]_{j+1,j} \quad (57)$$

$$\{FG\}_e = \left[\{DET\}^T[M3000] + \nu^1_{GJ}\{ETAKJ\}^T[M40K00]\{DET\}\right]\{G\}'_{j+1}$$
$$+ \frac{\Delta t}{2}\Big[-\{UBARK\}^T[M30K0](\{G\}+\{P\}) + \{ETAKJ\}^T[M30K0]\{QJ\}$$
$$+ \{ETAKJ\}^T[M40K00]\{UL\}\{SIGLJ\}$$
$$+ \nu^2_{GJ}\{ETAKJ\}^T[M40KL0]\{UBARL\}\{G\}\Big]_{j+1,j} \quad (58)$$

$$\{FP\}_e = \{DET\}^T[M3000]\{P\} - (\gamma-1)\Big[\{DET\}^T[M3000]\{G\}$$

$$+ \tfrac{1}{2}\{DET\}^T[M40000]\{UJ\}\{MJ\}\Big]_{j+1} \tag{59}$$

$$\{FSIJ\}_e = \{DET\}^T[M3000]\{SIGIJ\} - \frac{\mu}{Re}\Big[\{ETAKJ\}^T[M30K0]\{UI\}$$

$$+ \{ETAKI\}^T[M30K0]\{UJ\} - \tfrac{1}{3}\delta_{IJ}\{ETAKL\}^T[M30K0]\{UL\}\Big]_{j+1} \tag{60}$$

$$\{FQJ\}_e = \{DET\}^T[M3000]\{QJ\} - \kappa\Big[\{ETAKI\}^T[M30K0]\{GSR\} - \{UJUJ\}\Big]_{j+1} \tag{61}$$

$$\{FRK\}_e = \Big[\{DET\}^T[M3000] + \nu^1_{TJ}\{ETAKJ\}^T[M40K00]\{DET\}\Big]\{RK\}'_{j+1}$$

$$+ \frac{\Delta t}{2}\Big[-\{UBARK\}^T[M30K0]\{RK\} + \{DET\}^T[M3000]\{RE\}$$

$$+ \{ETAKJ\}^T[M400K0]\{UI\}\{SIJ\}$$

$$- \{ETAKJ\}^T((\{ETALI\}^T[M500K0L])\{KDIFIJ\})\{RK\}$$

$$+ \nu^2_{TJ}\{ETAKJ\}^T[M40KL0]\{UBARL\}\{RK\}\Big]_{j+1,j} \tag{62}$$

$$\{FRE\}_e = \Big[\{DET\}^T[M3000] + \nu^1_{EJ}\{ETAKJ\}^T[M40K00]\{DET\}\Big]\{RE\}'_{j+1}$$

$$+ \frac{\Delta t}{2}\Big[-\{UBARK\}^T[M30K0]\{RE\} + \{ESK\}^T[M40000]\{DET\}\{RE\}$$

$$+ \{ETAKJ\}^T[M400K0]\{UI\}\{SIJ\cdot ESK\}$$

$$- \{ETAKJ\}^T((\{ETAKI\}^T[M500K0L])\{EDIFIJ\})\{RE\}$$

$$+ \nu^2_{EJ}\{ETAKJ\}^T[M40KL0]\{UBARL\}\{RE\}\Big]_{j+1,j} \tag{63}$$

A few comments regarding notational structure are appropriate. The elemental hypermatrices [M...], for an n-dimensional problem description, are evaluated once and for all by integrals over a master finite element domain R_e^n. For a one-dimensional domain R^1, i.e., [A...], all matrices through [A4....] are listed in Appendix A, for both the linear (k=1) and quadratic (k=2) cardinal basis $\{N_k\}$. For two- and three-dimensional domains R^n, ie., [B...] and [C...], all element matrices through [B4...] and [C2...] are listed in Appendix B and C, respectively, for the bi-linear tensor product cardinal basis $\{N_1\}$. The discrete indices J, K, L, occurring in both matrix and variable (FORTRAN) names, are tensor summation indices with range $1 \leq (J,K,L) \leq n$. The multi-pole coefficient $\vec{\beta}_2$ is expressed in terms of Cartesian scalar compoennts $\nu_{\alpha J}^{\gamma}$, with distinct values for each dependent variable.

The arrays {DET} and {ETAKJ} contain nodal values of the determinant J and elements of $\partial\eta_k/\partial x_j$. The nodal values of the contravariant scalar components of the convection velocity $\bar{u}_k$, equation 39, are denoted {UBARK}. The elements of {SIGIJ} are nodal values of the stress tensor computed in principal coordinates in terms of $u_j = m_j/\rho$. Equation 60 is evaluated only for the laminar flow Stokes stress tensor definition, equation 4, and δ_{IJ} is the Kronecker delta. The additional terms resulting from inclusion of the Reynolds stress constitutive equation 9, see equation 8, are readily evaluated using the illustrated expansions. In equation 61, {GSR} denotes nodal values of $e \equiv g/\rho$, while {UIUI} contains nodal values of the specific kinetic energy $\frac{1}{2}u_i u_i$. For equations 62-63, {KDIFIJ} and {EDIFIJ} contain as elements the nodal values of the "effective diffusion" coefficient tensors, $C_k\overline{\rho u_i' u_j'}\ k/\varepsilon - \bar{\mu}\delta_{ij}$ and $C_\varepsilon\overline{\rho u_i' u_j'}\ k/\varepsilon$, respectively. Further in equation 63, the elements of {ESK} and {SIJ·ESK} are nodal values of $C_\varepsilon^2\varepsilon/k$ and $C_\varepsilon^1\overline{\rho u_i' u_j'}\varepsilon/k$. In all equations, $\{\cdot\}'_{j+1}$ denotes $\{\cdot\}^p_{j+1} - \{\cdot\}_j$. The notation $]_{j+1,j}$ defines evaluation of the argument at t_{j+1}, and at t_j, followed by addition after multiplication by Θ and $(1-\Theta)$, see equation 24.

5. Tensor Jacobian Matrix Statement

The algorithm matrix statement is completed by construction of the tensor matrix product Jacobian, see equations 27 and 42. Equations 56-63

provide the expressions for $\{FI\}_e$, which are analytically differentiable by members of $\{QJ\}_e$. The construction of certain terms involves differentiation with respect to the parameter $\bar{u}_k = \bar{m}_k/\rho$. Using the chain rule, and equations 38-39,

$$\frac{\partial}{\partial\{MI\}_e} = \frac{\partial}{\partial\{MI\}} + \frac{\partial}{\partial\{\bar{M}K\}}\frac{\partial\{\bar{M}K\}}{\partial\{MI\}} + \frac{\partial}{\partial\{\bar{U}K\}}\frac{\partial\{\bar{U}K\}}{\partial\{MI\}}$$

$$= \frac{\partial}{\partial\{MI\}} + \det J \frac{\partial\eta_k}{\partial x_i}\left[\frac{\partial}{\partial\{\bar{M}K\}} + \frac{1}{\bar{\rho}}\frac{\partial}{\partial\{\bar{U}K\}}\right] \qquad (64)$$

$$\frac{\partial}{\partial\{R\}_e} = \frac{\partial}{\partial\{R\}} - \left(\frac{\bar{m}_k}{\bar{\rho}^2}\right)\frac{\partial}{\partial\{\bar{U}K\}} \qquad (65)$$

Denote $\bar{D}_{KI}$ as the element average of $\det J(\partial\eta_k/\partial x_i)_e$, and define K to signify the discrete free index, ie., no summation implied, corresponding to $\partial/\partial\eta_k$, see equations 53-54. The non-empty tensor product Jacobians for equations 56-61, suppressing the subscript e throughout, are

$$[JRR]_e = \{DETK\}^T[A3000] + \nu^1_{RJ}\{ETAKJ\}^T[A40K00]\{DETK\}$$
$$+ \frac{\Delta t}{2}\nu^2_{RJ}\left[\{ETAKJ\}^T[M40KL0]\{UBARL\} - \left(\frac{\bar{m}_L}{\bar{\rho}^2}\right)\{ETAKJ\}^T[M40K0L]\{R\}\right]$$

$$[JRMI]_e = \frac{\Delta t}{2}\left[-\{ETAKI\}^T[A30K0] + \nu^2_{RJ}\bar{D}_{KI}\{ETAKJ\}^T[A30KK]\right] \qquad (66)$$

$$[JMIR]_e = \frac{\Delta t}{2}\left(\frac{\bar{m}_k}{\bar{\rho}^2}\right)\left[\{MI\}^T[A30K0] + \nu^2_{IJ}\{MI\}^T[A4KK00]\{ETAKJ\}\right]$$

$$[JMIMI]_e = \{DETK\}^T[A3000] + \nu^1_{IJ}\{ETAKJ\}^T[A40K00]\{DETK\}$$
$$+ \frac{\Delta t}{2}\left[-\{UBARK\}^T[A30K0] - \bar{D}_{KI}\{MI\}^T[A30K0]\frac{1}{\bar{\rho}}\right.$$
$$\left. + \nu^2_{IJ}(\{UBARK\}^T[A40KK0]\{ETAKJ\} + \frac{1}{\bar{\rho}}\{MI\}^T[A4KK00]\{ETAKJ\})\right]$$

$$[JMIMJ]_e = \frac{\Delta t}{2}\bar{D}_{KJ}\left(\frac{1}{\bar{\rho}}\right)\left[-\{MJ\}^T[A30K0] + \nu^2_{JL}\{MJ\}^T[A4KK00]\{ETAKL\}\right]$$

$$[JMIP]_e = -\frac{\Delta t}{2}\{ETAKI\}^T[A30K0]$$

$$[JMISIJ]_e = \frac{\Delta t}{2}\{ETAKJ\}^T[A40K00]\{DETK\} \tag{67}$$

$$[JGR]_e = \frac{\Delta t}{2}\left(\frac{\bar{m}_k}{\bar{\rho}^2}\right)\left[\{G+P\}^T[A30K0] + \nu_{GJ}^2\{G\}^T[A4KK00]\{ETAKJ\}\right]$$

$$[JGMI]_e = \frac{\Delta t}{2}\bar{D}_{KI}\left(\frac{1}{\bar{\rho}}\right)\left[-\{G+P\}^T[A30K0] + \nu_{GL}^2\{G\}^T[A4KK00]\{ETAKJ\}\right]$$

$$[JGG]_e = \{DETK\}^T[A3000] + \nu_{GJ}^1\{ETAKJ\}^T[A40K00]\{DETK\}$$

$$+ \frac{\Delta t}{2}\left[-\{UBARK\}^T[A30K0] + \nu_{GJ}^2\{UBARK\}^T[A40KK0]\{ETAKJ\}\right]$$

$$[JGP]_e = -\frac{\Delta t}{2}\{UBARK\}^T[A30K0]$$

$$[JGSIJ]_e = \frac{\Delta t}{2}\{ETAKJ\}^T[A40K00]\{UI\}$$

$$[JGQJ]_e = \frac{\Delta t}{2}\bar{D}_{LJ}\{ETAKL\}^T[A30K0] \tag{68}$$

$$[JPR]_e = -\left(\frac{\gamma-1}{2}\right)\left(\frac{\bar{m}_k}{\bar{\rho}^2}\right)\{DETK\}^T([A40000]\{MK\}$$

$$[JPMK]_e = \left(\frac{\gamma-1}{2}\right)\left[\{DETK\}^T([A40000]\{UK\}) + \left(\frac{1}{\bar{\rho}}\right)\{DETK\}^T([A40000]\{MK\})\right]$$

$$[JPG]_e = -(\gamma-1)\{DETK\}^T[A3000]$$

$$[JPP]_e = \{DETK\}^T[A3000] \tag{69}$$

$$[JSIJR]_e = \frac{\mu}{\bar{\rho}^2}\left[\bar{m}_i\{ETAKJ\}^T[A30K0] + \bar{m}_j\{ETAKI\}^T[A30K0]\right.$$

$$\left. - \tfrac{1}{3}\delta_{IJ}\bar{m}_\ell\{ETAKL\}^T[A30K0]\right]$$

$$[JSIJMI]_e = -\frac{\mu}{\bar{\rho}}\left[\{ETAKJ\}^T[A30KO] - \tfrac{1}{3}\delta_{IJ}\{ETAKL\}^T[A30KO]\right]$$

$$[JSIJSIJ]_e = \{DETK\}^T[A3000] \tag{70}$$

$$[JQIR]_e = \frac{\kappa}{\bar{\rho}^2}\left[\bar{g}\{ETAKI\}^T[A30KO] - \frac{\bar{m}_i^2}{\bar{\rho}}\{ETAKI\}^T[A30KO]\right]$$

$$[JQIMI]_e = \frac{\kappa\bar{m}_i}{\bar{\rho}^2}\{ETAKI\}^T[A30KO]$$

$$[JQIG]_e = -\frac{\kappa}{\bar{\rho}}\{ETAKI\}^T[A30KO]$$

$$[JQIQI]_e = \{DETK\}^T[A3000] \tag{71}$$

The scalars $\bar{m}_i$, $\bar{\rho}$ and $\bar{g}$ are element average values of {MI}, {R} and {G} respectively. In the formation of equations 66-71, the Boolean index K(=1) in various A-matrices has been permuted to facilitate differentiation of each expression by the last right contraction matrix. There is considerable commonality pervading the individual element Jacobian constructions. The elements of each Jacobian are formed on the finite element domain R_e^1, using the element matrices listed in Appendix A, and then assembled into the global form using the operator S_e, see equation 55. In actual practice, the column matrix {δQI} is ordered on degrees of freedom at a node, e.g., $\{\ldots, \delta R_j, \delta U_j, \delta E_j, \delta P_j, \delta R_{j+1}, \ldots\}^T$. Hence, the global tensor Jacobian $[J_\eta]$, see equation 42, is α-block tri-diagonal using the linear (k=1) finite element basis, and α-block penta-diagonal for the quadratic (k=2) cardinal basis.

6. Comments on the Algorithm

Equations 56-63 are an exact statement of the elemental matrix equivalent of the finite element algorithm statement, equation 24, which upon assembly on UR_e^n yields equation 24. While the master hypermatrix symbology $[MP...]_e$ is independent of the completeness of the approximation subspace, ie., the degree k of $\{N_k(\vec{\eta})\}$, equation 21, the order of each square matrix is equal to $(k+1)^n$. In addition, since each master matrix is a hypermatrix of degree one at least, $P \geq 1$, there are $(k+1)^P$ entries for each element of each matrix $[MP...]_e$.

The associated inner product DO-loop to form {FI}, especially for $k > 1$, would be of excessive length on a scalar machine. The algorithm matrix statement can be simplified in this regard, with the commission of interpolation error only. Each Boolean index "0" appearing in a hypermatrix $[MP...]_e$ signifies the matrix inner product with the element-dependent column (row) matrix $\{\cdot\}_e$ corresponding to an interpolation on R_e^n. For example, the lead term in each of equations 56-63 is

$$\{DET\}_e^T[M3000]_e$$

In the instance of the linear transformation $\vec{x} = \vec{\eta}$, the elements of $\{DET\}_e$ are each equal to a constant times the element measure. Hence, $\{DET\}_e^T = (DET)_e\{ONE\}^T$, where the elements of {ONE} are unity, which yields

$$\begin{aligned}\{DET\}_e^T[M3000] &= (DET)_e\{ONE\}^T[M3000] \\ &= (DET)_e[M200] \end{aligned} \qquad (72)$$

where $(DET)_e$ is the scalar measure (area, volume) of the finite element domain R_e^n. As presented in Section V, for k=1, n=2, and a general quadrilateral domain R_e^n, the elements of $\{DET\}_e$ are,

$$\{DET\}_e = \frac{1}{4} \begin{Bmatrix} (X2-X1)(Y4-Y1) - (X4-X1)(Y2-Y1) \\ (X2-X1)(Y3-Y2) - (X3-X2)(Y2-Y1) \\ (X3-X4)(Y3-Y2) - (X3-X2)(Y3-Y4) \\ (X3-X4)(Y4-Y1) - (X4-X1)(Y3-Y4) \end{Bmatrix}_e \tag{73}$$

with counterclockwise node numbering starting in the lower left corner of R_e^2, see Figure 1a). Defining the sum of the entries as $(\overline{DET})_e$, and for a "decent" element aspect ratio, equation 72 can be approximated as

$$\{DET\}_e^T[B3000] \approx (\overline{DET})_e[B200] \tag{74}$$

with the commission of interpolation error only. This operation reduces by $(k+1)^P$ the number of multiplications to form this term in {FI}.

In this context then, every hypermatrix inner-product statement in equations 56-63 is eligible for reduction of hypermatrix degree through elemental averaging. In terms of error, this is appropriate only for element-dependent data that are sufficiently smooth. In general, this rules out the averaging of any dependent variable, since non-smooth solutions must be admitted. However, for "decent" discretizations UR_e^n, the geometric data should be eligible for averaging on sufficiently refined grids. Hence, for example in equation 57,

$$\{ETAKJ\}_e^T[M40K00]\{DET\}_e \rightarrow (\overline{ETAKJ})_e(\overline{DET})_e[M2K0]$$

$$\{ETAKJ\}_e^T[M40KL0]\{UBARL\}_e \rightarrow (\overline{ETAKJ})_e\{UBARL\}_e^T[M30KL] \tag{75}$$

In equation 62, for example

$$\{ETAKJ\}_e^T((\{ETALI\}_e^T[M500K0L])\{KDIFIJ\}_e)$$

$$\rightarrow (\overline{ETAKJ})_e(\overline{ETAKL})_e\{KDIFIJ\}_e^T[M30KL] \tag{76}$$

Hence, no hypermatrix of degree $P > 1$ is required formed or stored, provided the discretization of R_e^n is of sufficient quality. The averaging of the grid data in the Jacobian matrix formulation, equations 66-71, is thus also appropriate.

A second point to note is that definition and use of the stress tensor σ_{ij}, and heat flux vector q_j, as dependent variables has permitted algorithm formulation without evaluation of second order spatial derivatives in the generalized coordinates framework. Hence, the troublesome, sometimes destabilizing mixed derivatives resulting from viscosity terms in the direct (laminar flow) formulation, are totally absent. The penalty for the dependent variable construction is a significant increase in size of the block-banded tensor product Jacobian. Conversely, a general turbulent flow prediction can ostensibly be handled with ease, by expansion of the defining equation for σ_{ij} and q_j. The fact that the algebraic "constitutive" equations for p, σ_{ij} and q_j are handled directly within the weighted residuals algorithm lends an overall uniformity that simplifies construction.

Finally, non-iterative and direct steady-state forms of the algorithm are special cases of the formulation. The non-iterative construction simply constitutes acceptance of the first solution $\{\delta QI\}^2_{j+1}$ of the Newton algorithm using only evaluation of $\{FI\}^1_{j+1}$, see equation 25. Since the iteration index $p > 0$ by definition, $\{QI\}^1_{j+1} \equiv \{QI\}_j$. Therefore, $\{QI\}'_{j+1} = \{0\}$ in equations 56-63, and the corresponding expressions in brackets are not evaluated. Furthermore, $\Delta t/2 \rightarrow \Delta t$ and the evaluation $\cdot]_{j+1,j}$ reduces to $\cdot]_j$. Since the terms in $\{FI\}$ involving $\nu^1_{\alpha j}$ have been eliminated, so are the corresponding terms in the tensor product Jacobians, equations 66-68. The multiplier $\Delta t/2$ remains appropriate in $[J_\eta]$, and evaluations are made using $\{QI\}_{j+1} \equiv \{QI\}_j$. This procedure thus reduces the algorithm operations count by a significant factor, at the expense of removal of $\nu^1_{\alpha j}$ from the construction.

The direct steady-state algorithm is identical to the non-iterative formulation except that $\Delta t/2 \rightarrow \Delta t$ in the Jacobians. This multiplier, common to all elements of $\{FI\}$, can be divided out, yielding Δt^{-1} as a scalar multiplier on $\{DET\}^T[A3000]$ in the self coupling Jacobians [JQQ] for the initial-value dependent variables. This yields the tensor matrix generalization of the "approximate factorization" procedures devised for finite difference algorithms, cf. [5,6].

SECTION IV

THEORETICAL ANALYSIS, ACCURACY AND CONVERGENCE

1. General Concepts

The theoretical analysis of accuracy and convergence aspects of a finite element algorithm is typically quantized in inequalities in Sobolev norms. The Navier-Stokes problem class for finite Reynolds number is non-linear elliptic with initial-value character. For infinite Reynolds number (inviscid flow), the resulting (Euler) equation set is non-linear hyperbolic. The theoretical analyses are exact only for the linear, one-dimensional initial-value problem description, and the linear elliptic multi-dimensional problem class, cf. [11]. The convergence statements are constructed by quantizing the total approximation error in terms of component parts as:

$$\begin{aligned} &\text{semi-discrete approximation error:} && \varepsilon^h(x,n\Delta t) = q(x,n\Delta t) - q^h(x,n\Delta t) \\ &\text{discrete approximation error:} && \sigma(x,n\Delta t) = q(x,n\Delta t) - Q(x,n\Delta t) \\ &\text{temporal truncation error:} && \tau(x,n\Delta t) = Q(x,n\Delta t) - q^h(x,n\Delta t) \end{aligned} \quad (77)$$

For any choice of norm, the triangle inequality yields

$$\|\varepsilon\| = \|\sigma + \tau\| \leq \|\sigma\| + \|\tau\| \quad (78)$$

The semi-discrete approximation error in a finite element solution of a linear parabolic equation, using the fully explicit Euler integration algorithm, $\Theta=0$ in equation 24, satisfies the inequality [11, p. 408].

$$\|\varepsilon^h(n\Delta t)\|_{H^m(R^1)} \leq C_1\Delta_e^{(k+1-m)}\|q(n\Delta t)\|_{H^{k+1}(R^1)} + C_2\Delta t\|Q_o\|_{H^m(R^1)} \quad (79)$$

The C_α are constants independent of Δ_e, the measure of the uniform finite element discretization of R^1, $m=1$ for the parabolic specification, and $\|Q_o\|_{H^m}$ is the H^m norm of the interpolation of the initial data.

The corresponding statement for smooth solutions of a linear hyperbolic equation, and using explicit Euler integration, is [11, p. 418]

$$\|\varepsilon^h(n\Delta t)\| \leq C_1\Delta_e^{k+1}\|q(n\Delta t)\|_{H^{k+1}(R^1)} + C_2\Delta t\|Q_0\|_{H^0} + C_3\Delta_e\int_0^{n\Delta t}\|q(\tau)\|_{H^{k+1}}d\tau \qquad (80)$$

The C_α remain constants independent of Δ_e, the first term is the familiar bound on discretization refinement, and the second term is the H^0 norm of the initial-data interpolation. The third term is an additional discretization-sensitive error generated through propagation of the $k+1^{\underline{st}}$ derivative of the exact solution over $n\Delta t$. The exponent of unity on Δ_e is always smaller than the exponent k+1 on the lead term. Therefore, enhanced convergence through use of a more complete basis, $k > 1$ in equation 21 is not expected.

The convergence proof for a linear elliptic boundary value problem definition, on a multi-dimensional domain, can be written as [11]

$$\|\varepsilon^h\|_{H^\alpha(R^n)} \leq C_1\Delta^\gamma\|g\|_{H^p(\partial R)} \qquad (81)$$

where $\|g\|_{H^p}$ is the H^p Sobolev norm of the boundary data specification of the problem. C_1 remains a constant independent of the extremum mesh measure Δ, $\alpha \leq 1$, and $\gamma = \min[k, p + s - \frac{1}{2}, 1 - \alpha]$ for $p \geq \frac{1}{2}$ and $s \geq \frac{1}{2}$. The parameters p and s pertain to the regularity of the discretization of UR_e^n, and convergence can become independent of the degree k of $\{N_k\}$ with distorted meshes having large discretization aspect ratios. Otherwise, γ takes on the value k.

In regards to accuracy, an important additional theoretical aspect is extremization of the (semi-) discrete approximation error by a finite element solution. Strang and Fix [24] prove, for the one-dimensional Laplacian, that the corresponding linear (k=1) finite element solution produces an absolute minimum discrete approximation error $\varepsilon^h = q - q^h$, measured in the energy norm $E(\varepsilon^h,\varepsilon^h)$, which is defined as

$$E(q^h,q^h) \equiv \int_{R^1} \frac{dq^h}{dx} \frac{dq^h}{dx} dx \tag{82}$$

The theorem states,

$$E(q-q^h,\ q-q^h) \leq E(q-\bar{q},\ q-\bar{q}) \tag{83}$$

for all possible choices of an alternative approximation $\bar{q}$, including the (linear) interpolation of the exact solution and/or a finite difference solution. Generalizing to degree k, and using orthogonality, they further prove that the energy in the approximation error is bounded in the form

$$E(\varepsilon^h,\varepsilon^h) \leq C_1 \Delta^{2k} \|q\|^2_{H^{k+1}} \tag{84}$$

Again, C_1 is a constant independent of the mesh, Δ is the extremum measure, and $\|q\|^2_{H^{k+1}}$ is the square of the k+1 Sobolev norm of the exact solution.

$$\|q\|_{H^{k+1}} \equiv \int_{R^1} \left[q^2 + \left(\frac{dq}{dx}\right)^2 + \ldots + \left(\frac{d^{k+1}q}{dx^{k+1}}\right)^2 \right]^{\frac{1}{2}} dx \tag{85}$$

The statement corresponding to equation 81 is then,

$$E(\varepsilon^h,\varepsilon^h) \leq C_2 \Delta^{2k} \|g\|^2_{H^0} \tag{86}$$

where $\|g\|_{H^0}$ is the H^0 Sobolev norm of the boundary data, which need only be square integrable.

In approaching the Navier-Stokes problem class, Baker and co-workers document results of detailed computational experiments, defined to confirm the convergence exponent 2k, equation 86, or k in the equation 79, for select model equation systems. For a linear, transient parabolic heat conduction equation [25], the convergence data were exactly interpolated by k over the range $1 \leq k \leq 3$. This confirmation was similarly obtained [26] for $1 \leq k \leq 2$, for solution of the mildly non-linear laminar flow boundary layer equation system, for a range of pressure gradients. Significantly, the energy in the

semi-discrete, k=1 approximation error $E(\varepsilon^h,\varepsilon^h)$, equation 86, was confirmed extremum in comparison to the Crank-Nicolson algorithm solution, including use of non-uniform discretizations. Confirming results, on error extremization and convergence rate exponent for $1 \leq k \leq 2$, are reported [27] for the highly non-linear turbulent boundary layer equation system solution. Of principal additional significance, the energy norm was a strongly nonlinear function of the turbulent "eddy viscosity," yet the semi-discrete error convergence data were closely interpolated by lines of constant slope equal to 2k. Finally, for a smooth solution to a linear hyperbolic equation [25], the convergence rate was indeed independent of k, confirming dominance of the third term in equation 80. In all cited cases, on a sufficiently refined discretization, the absolute level of semi-discrete approximation error for the $k > 1$ algorithm solutions was uniformly less than that for the k=1 solutions. Furthermore, in comparison to the equal complexity Crank-Nicolson finite difference algorithm, the k=1 finite element semi-discrete approximation error in $\|\cdot\|_{H^1}$ was uniformly mimimum for any discretization of R^1.

2. The Parameter Set $\vec{\beta}_2$

A principal requirement for high Mach number Navier-Stokes or Euler solutions is prediction of non-smooth solutions, ie., ones exhibiting a finite number of finite discontinuities (shocks). The finite element theoretical literature does not contain convergence statements analygous to equations 79-80, nor is extremization of the semi-discrete approximation error guaranteed. The theoretical requirement is to estimate a reasonable value for the multi-pole expansion coefficient set $\vec{\beta}_2$, equation 22. For the one-dimensional form of the continuity equation 1, and for an imposed constant convection velocity $u_j \equiv U_0$, the exact Fourier solution is,

$$q(x,t) = Q_0 \exp[i\omega(x-U_0 t)] \tag{87}$$

corresponding to propagation of the initial data Q_0 with group velocity U_g.

The semi-discrete Fourier solution is,

$$q^h(x,t) = Q_0 \exp\,[i\omega(j\Delta x - \Gamma U_0 t)] \tag{88}$$

where $\Gamma \equiv \sigma + i\delta$, where σ and δ are real numbers, $i = \sqrt{-1}$, $\omega = 2\pi/\lambda$ is the wave number for Fourier mode of wavelength λ, and $x = j\Delta x$, $j = 0,1,2,\ldots,$ is represented by discrete intervals of (uniform) measure Δx. Raymond and Garder [28], for the definition $\vec{\beta}_2 \equiv \nu\Delta x\hat{i}$, where $\nu > 0$ is a scalar parameter, determined the k=1 algorithm functional forms for σ and δ as,

$$\sigma = 1 + \left(\frac{-1}{180} + \frac{\nu^2}{12}\right)d^4 + O(d^6) \tag{89}$$

$$\delta = -\frac{\nu}{12}\,d^3 + O(d^5) \tag{90}$$

In equations 89-90, $d \equiv \omega\Delta x = 2\pi/n$, where n is the discrete Fourier mode index, $\lambda_n = n\Delta x$, and O indicates order. Hence, the semi-discrete Fourier solution q^h can be made a sixth-order accurate approximation to equation 87 by eliminating the $O(d^4)$ term in equation 89, yielding $\nu \equiv (15)^{-\frac{1}{2}}$. Correspondingly, $\delta < 0$ in equation 90, and an artificial dissipation mechanism becomes introduced, yielding the semi-discrete approximation as

$$q^h(x,t) = Q_0\exp[i\omega(j\Delta x - U_0 t)]\exp[-\omega^4 kt + O(\Delta x^6)] \tag{91}$$

Here, $k \equiv U_0(\Delta x^3)/12\sqrt{15}$ is the coefficient of "artificial viscosity," and the dissipation mechanism is highly phase selective due to the ω^4 factor.

The original analysis was expanded [15] by redefining the dissipation parameter $\vec{\beta}_2$ in the form,

$$\vec{\beta}_2 \equiv \Delta x[\nu^1\delta_t + \nu^2\delta_x]\hat{i} \tag{91}$$

where δ_t and δ_x are Kronecker delta-type functions, yielding ν^1 operating on the time derivative, and ν^2 operating on the spatial derivative term, only, in the one-dimensional form of equation 1. Proceeding through the substitutions yields, for the k=1 algorithm.

$$\sigma = 1 - d^2(\nu^1 - \nu^2) + d^4\left[\frac{1}{180} + \frac{\nu^1\nu^2}{12} + (\nu^1 - \nu^2)(\nu^1)^3\right] + O(d^6) \qquad (92)$$

$$\delta = d\left[\nu^1 - \nu^2\right] - d^3\left[\frac{\nu^2}{12} - (\nu^1 - \nu^2)(\nu^1)^2\right] + O(d^5) \qquad (93)$$

For the quadratic (k=2) algorithm, the form of equation 92 is [15].

$$\sigma = 1 - 4\nu^1\nu^2 + d^2\left[-14 + 184\nu^1\nu^2 - 60(\nu^1)^2 + 240\nu^2(\nu^1)^3\right]/15$$
$$+ d^4\left[\frac{2}{5} - \frac{542}{150}\nu^1\nu^2 + \frac{8}{3}(\nu^1)^2 - \frac{672}{15}\nu^2(\nu^1)^3\right.$$
$$\left. + 16(\nu^1)^4 - 64\nu^2(\nu^1)^5\right] + O(d^6) \qquad (94)$$

Setting $\nu^1 \equiv \nu \equiv \nu^2$ in equations 92-93 yields the results of the original analysis [28]. Enforcing sixth-order accuracy for the k=1 algorithm construction, equation 92, yields the constraint

$$\nu^2 = \frac{d^2\left[\frac{1}{180} - (\nu^1)^4\right] + (\nu^1)^2}{d^2\left[\frac{\nu^1}{12} - (\nu^1)^3\right] + \nu^1} \qquad (95)$$

Figure 2 is a plot of equation 95 with n as a parameter. Sixth-order accuracy can be achieved only for $\nu^1 > 0$; for any level, ν^2 ranges over an order of magnitude dependent upon n, with the largest levels required for the shortest wave lengths. All curves converge to the point $\nu^1 = (15)^{-\frac{1}{2}} = \nu^2$. Defining $\nu^1 \equiv 0$ renders both the k=1 and k=2 algorithm construction for the semi-discrete approximation q^h, equation 91, a second-order accurate representation of the analytical solution.

The phase velocity of each Fourier mode can be determined for the fully discrete solution Q, defined as

$$Q \equiv q_n^h(\Delta x, n\Delta t) = g^n \exp[i\omega j\Delta x] \qquad (96)$$

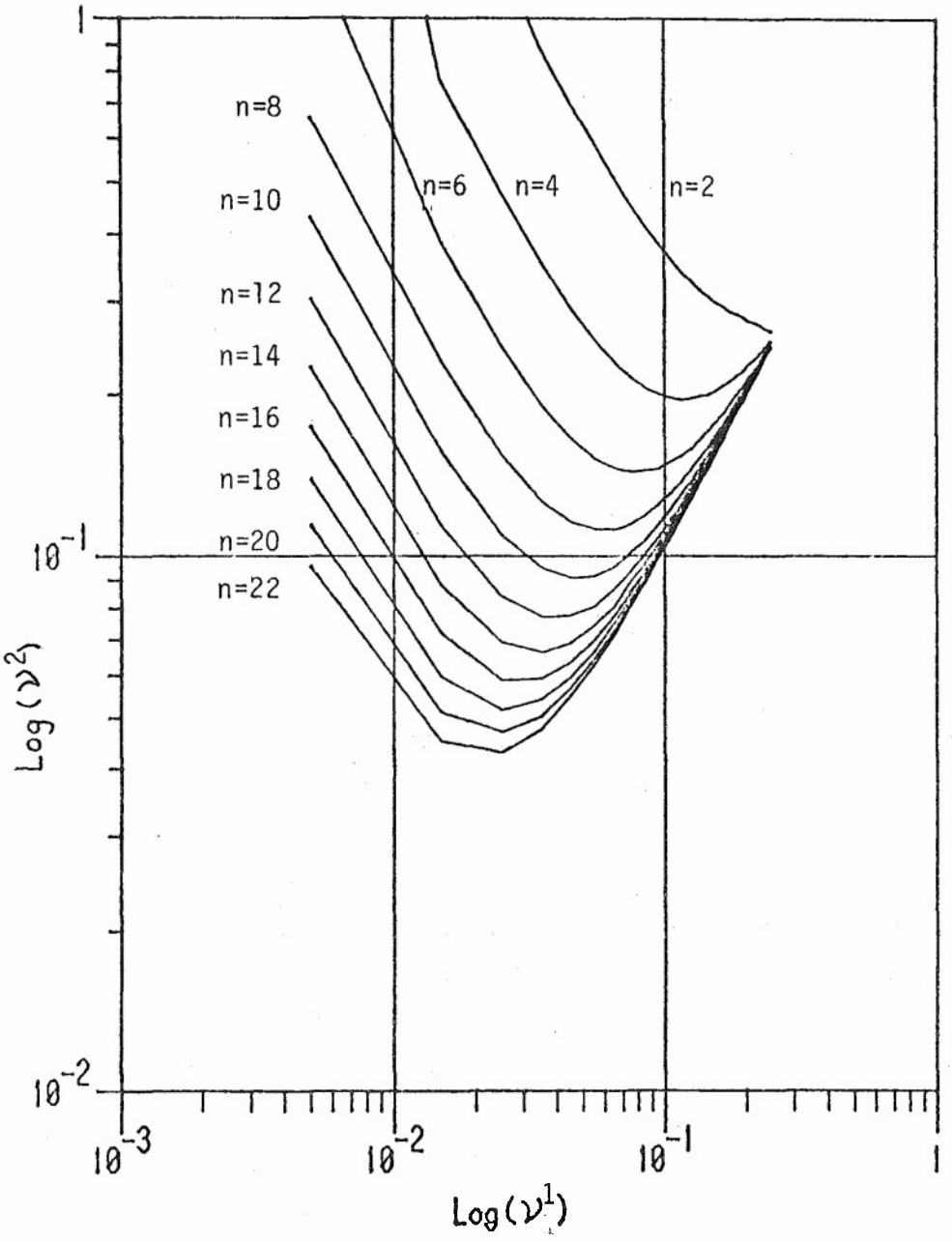


Figure 2. Distribution of Solution to Equation 95 For ν Optimal.

Substituting equation 96 into the one-dimensional continuity equation, inserted into the algorithm statement, equation 22, integrating using the implicit trapezoidal rule, $\Theta = \frac{1}{2}$ in equation 24, and defining $\vec{\beta}_2 \equiv \nu\Delta x\hat{i}$, yields the amplification factor in equation 96 in the form [29].

$$g = \frac{1+\frac{1}{2}\cos(\omega\Delta x) - 3C\nu\sin^2(\frac{1}{2}\omega\Delta x) - i\frac{3}{4}(C+2\nu)\sin(\omega\Delta x)}{1+\frac{1}{2}\cos(\omega\Delta x) + 3C\nu\sin^2(\frac{1}{2}\omega\Delta x) + i\frac{3}{4}(C=2\nu)\sin(\omega\Delta x)} \qquad (97)$$

In equation 97, $C \equiv U_0\Delta x/\Delta t$ is the non-dimensional time step (Courant number), and ν is the dissipation parameter. For $\nu \equiv 0$, $|g| = 1$ for all Δt and Δx, indicating the k=1 finite element algorithm will neither damp nor amplify solution disturbances, ie., it is neutrally stable. For all $\nu > 0$, $|g| < 1$ and an artificial dissipation mechanism becomes introduced.

The non-dimensional phase velocity Θ_n of each discrete Fourier mode is defined in terms of the real and imaginary components of g as [29]

$$\Theta_n \equiv \frac{1}{\lambda}\tan^{-1}\left[\frac{\text{imag}(g)}{\text{real}(g)}\right] \qquad (98)$$

Figure 3 graphs equation 98 for select levels of C and ν, as a function of the discrete wavelength parameter n, recall $\lambda_n = n\Delta x$. The exact solution is $\Theta_n \equiv 1$ for all n. The fully discrete solution cannot resolve the "$2\Delta x$" mode, n=2. For $C \leq 0.5$ and $\nu = 0$, $\Theta_n \geq 0.95$ for all $n \geq 5$, ie., the discrete solution will artificially disperse these low-wave length solution modes yielding solution "wiggles," cf. [17]. Increasing the Courant number further retards the phase velocities for all n. Setting $\nu = (15)^{-\frac{1}{2}}$ and $C = 0.5$, yields $\Theta_n \geq 0.97$ for all $n > 2$, which essentially eliminates the short wave length phase dispersion. Unfortunately, this excellent phase accuracy has associated with it a high level of artificial dissipation which excessively diffuses the gradients in the physical solution. For comparison, the dashed curves in Figure 3 interpolate the data for Θ_n as obtained for the Crank-Nicolson second order accurate, $\Theta = \frac{1}{2}$ finite difference solution algorithm applied to the one-dimensional continuity equation with U_0. Unacceptable phase dispersion is indicated for all $n < 20$.

The multi-dimensional, multi-dependent variable, generalized coordinates definition for the dissipation parameter $\vec{\beta}_2^{\alpha}$ is

$$\vec{\beta}_2^{\alpha} \equiv (\det J)[\vec{\nu}_{\alpha}^{1}\ \delta_t + \vec{\nu}_{\alpha}^{2}\ \delta_x] \tag{99}$$

In equation 99, det J is the measure of Δ_e^n, subscript α denotes the appropriate member of q_{α}^{h}, and the parameter vectors $\vec{\nu}_{\alpha}^{K}$ are expressed in terms of scalar components $\nu_{\alpha j}^{K}$ in the x_j coordinate system, see equations 56-63.

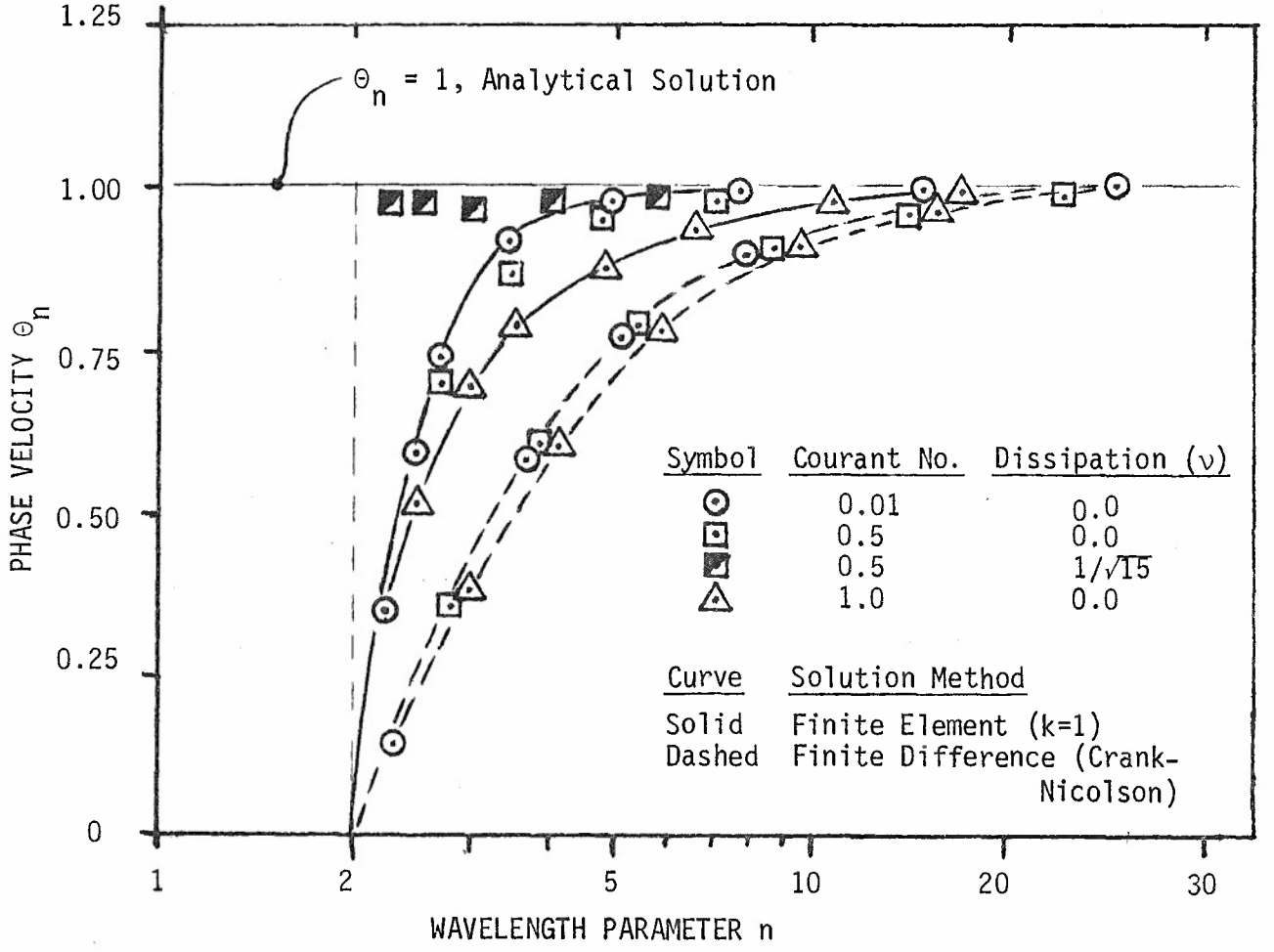


Figure 3. Fourier Phase Velocity Distribution for Finite Element and Finite Difference Algorithms.

SECTION V.

RESULTS AND DISCUSSION

1. Quasi-One Dimensional Flow

Several essential aspects of algorithm performance regarding accuracy, convergence, etc., can be quantized by examination of quasi-one-dimensional inviscid flows. Defining the convection velocity $u \equiv m/\rho$, and the flow cross-sectional area as $A(x)$, the governing differential equation set is formed from equations 1-7 as

$$L(\rho) = \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}[m] + \rho u \frac{\partial \ell n A}{dx} = 0 \tag{100}$$

$$L(m) = \frac{\partial m}{\partial t} + \frac{\partial}{\partial x}[um + p] + mu \frac{\partial \ell n A}{\partial x} = 0 \tag{101}$$

$$L(g) = \frac{\partial g}{\partial t} + \frac{\partial}{\partial x}[ug + up] + u(g+p) \frac{\partial \ell n A}{\partial x} = 0 \tag{102}$$

$$L(p) = p - (\gamma-1)[g - \tfrac{1}{2}um] = 0 \tag{103}$$

The finite element algorithm statement, equations 56-59, requires the metric data $\{DET\}_e$ and $\{ETAKJ\}_e$. For the affine coordinate transformation $x_1 = \eta_1$, and with the origin of the η_1 coordinate at the element centroid, cf. Figure 1, the elements of the cardinal bases $\{N_k(\eta)\}$, $1 \leq k \leq 2$, are

$$\{N_1(\eta)\} = \tfrac{1}{2}\begin{Bmatrix} 1 - \eta \\ 1 + \eta \end{Bmatrix} \tag{104}$$

$$\{N_2(\eta)\} = \tfrac{1}{2}\begin{Bmatrix} -\eta(1-\eta) \\ 2(1-\eta)(1+\eta) \\ \eta(1+\eta) \end{Bmatrix} \tag{105}$$

By definition, $x_1 \equiv \{N_k(\eta)\}^T\{XI\}_e$, and det J = det$[\partial x_1/\partial \eta_1]$. Denoting the elements of $\{XI\}_e$ as the left (L), right (R) and middle (M) x_1-coordinates, where $M \equiv \frac{1}{2}(R+L)$, then

$$k=1: \quad DET_e = \tfrac{1}{2}(R-L)$$

$$k=2: \quad DET_e = \tfrac{1}{2}(R-L) - \frac{3\eta}{4}(R+L) \tag{106}$$

Defining the element measure as $\Delta_e \equiv R-L$, then $\overline{DET}_e = \Delta_e/2$ for both the linear and quadratic basis. By the same procedures, the sole non-vanishing element of $\{ETAKJ\}_e$ is in the (1,1) location with a value of unity.

Therefore, the finite element algorithm statement becomes quite simplified for the quasi-one-dimensional situation. The specific form for equations 56-59 is

$$\{FR\}_e = \left[\Delta_e[A200] + \nu_1^1[A210]\right]\{R\}'_{j+1}$$
$$+ \frac{\Delta t}{2}\Big[[A210]\{M\} + \nu_1^2\{U\}^T[A3011]\{R\}$$
$$+ \{A\}^T([A41000]\{U\})\{R\}\Big]_{j+1,j} = \{0\} \tag{107}$$

$$\{FM\}_e = \left[\Delta_e[A200] + \nu_2^1[A210]\right)\{M\}'_{j+1}$$
$$+ \frac{\Delta t}{2}\Big[\{V\}^T[A3010]\{M\} + [A210]\{P\} + \nu_2^2\{U\}^T[A3011]\{M\}$$
$$+ \{A\}^T([A41000]\{U\})\{M\}\Big]_{j+1,j} = \{0\} \tag{108}$$

$$\{FG\}_e = \left[\Delta_e\ [A200] + \nu_3^1[A210]\ \right]\{G\}'_{j+1}$$
$$+ \frac{\Delta t}{2}\Big[\{U\}^T[A3010]\Big(\{G\} + \{P\}\Big) + \nu_3^2\{U\}^T[A3011]\{G\}$$

$$+ \{A\}^T([A41000]\{U\})(\{G\}+\{P\})\Big]_{j+1,j} = \{0\} \quad (109)$$

$$\{FP\}_e = \Delta_e[A200]\{P\} - \Delta_e(\gamma-1)\left[[A200]\{G\} - \frac{1}{2}\{U\}^T[A3000]\{M\}\right] = \{0\} \quad (110)$$

The Jacobian contributions for the Newton algorithm are, recall equations 66-69.

$$\frac{\partial\{FR\}}{\partial\{R\}} \equiv [JRR] = \Delta_e[A200] + \nu_1^1[A210] + \frac{\Delta t}{2}\left[\nu_1^2\left(\{U\}^T[A3011] - \frac{\bar{m}}{\bar{\rho}^2}\{R\}^T[A3110]\right) + \{A\}^T[A41000]\left(\{U\} - \frac{\bar{m}}{\bar{\rho}^2}\{R\}\right)\right]$$

$$[JRM] = \frac{\Delta t}{2}\left[[A210] + \nu_1^2[A211] + \{A\}^T[A3100]\right]_{j+1}^{p}$$

$$[JRG] = 0 = [JRP] \quad (111)$$

$$\frac{\partial\{FM\}}{\partial\{R\}} \equiv [JMR] = -\frac{\Delta t}{2}\left(\frac{\bar{m}}{\rho^2}\right)\{M\}^T\left[[A3010] + \nu_2^2[A3110] + [A40001]\{A\}\right]_{j+1}^{p}$$

$$[JMM] = \Delta_e[A200] + \nu_2^1[A210] + \frac{\Delta t}{2}\{U\}^T\left[[A3010] + \nu_3^2[A3011] + [A40001]\{A\}\right]_{j+1}^{p} + \frac{\Delta t}{2}\left(\frac{1}{\bar{\rho}}\right)\{M\}^T\left[[A3010] + \nu_2^2[A3110] + [A40001]\{A\}\right]_{j+1}^{p}$$

$$[JMG] = [0]$$

$$[JMP] = \frac{\Delta t}{2}[A210] \quad (112)$$

$$\frac{\partial\{FG\}}{\partial\{R\}} \equiv [JGR] = -\frac{\Delta t}{2}\left(\frac{\bar{m}}{\bar{\rho}^2}\right)\left[\{G\}^T([A3010] + \nu_3^2[A3110] + [A40001]\{A\}) + \{P\}^T[A3010]\right]_{j+1}^{p}$$

$$[JGM] = \frac{\Delta t}{2}\left(\frac{1}{\bar{\rho}}\right)\Big[\{G\}^T([A3010] + \nu_3^2[A3110] + [A40001]\{A\}) + \{P\}^T[A3010]\Big]_{j+1}^{p}$$

$$[JGG] = \Delta_e[A200] + \nu_3^1[A210] + \frac{\Delta t}{2}\{U\}^T\Big[[A3010] + \nu_3^2[A3011] + [A40001]\{A\}\Big]_{j+1}^{p}$$

$$[JGP] = \frac{\Delta t}{2}\{U\}^T\Big[[A3010] + [A40001]\{A\}\Big]_{j+1}^{p} \qquad (113)$$

$$\frac{1}{\Delta_e}\frac{\partial\{FP\}}{\partial\{R\}} \equiv [JPR] = -\left(\frac{\gamma-1}{2}\right)\left[\left(\frac{\bar{m}}{\bar{\rho}^2}\right)\{M\}^T[A3000]\right]_{j+1}^{p}$$

$$[JPM] = \frac{\gamma-1}{2}\Big[\{U\}^T[A3000] + \left(\frac{1}{\bar{\rho}}\right)\{M\}^T[A3000]\Big]_{j+1}^{p}$$

$$[JPG] = -(\gamma-1)[A200]$$

$$[JPP] = [A200] \qquad (114)$$

In equations 112-114, the superscript bar on m and ρ indicate the element average values. The elements of $\{A\}_e$ are the nodal values of ℓn A(x). The defined standard matrices and hypermatrices are listed in Appendix A for both cardinal basis formulations, $1 \leq k \leq 2$.

2. Accuracy and Convergence, Riemann Shock Tube

The principal requirement of any (quasi-) one-dimensional solution is to quantize the accuracy and convergence performance of the finite element algorithm. Mixed subsonic-supersonic flows are particularly useful, since shock sharpness and associated undershoot/overshoot are readily assessable. The Riemann shock tube problem, cf. [30, p. 1007], is well suited to this

analysis, since the resultant flow structure is richly endowed with discontinuities and sharp variable gradients, interspersed with planar plateau regions. The interesting case, with unique stagnation sound speeds in the two chambers separated by the diaphragm, has been exhaustively examined for algorithm performance in the finite difference field, cf. Van Lear [31] and Zalesak [32]. In particular, Sod [33] compares the results of approximately a dozen finite difference algorithms, based upon Lagrangian and mixed Lagrangian-Eulerian frameworks, for one shock tube specification.

The developed finite element algorithm is strictly Eulerian, and therefore not at all fine-tuned for the non-linear hyperbolic inviscid Riemann problem. Therefore, this problem serves the required purpose, as well as affording an interesting exact solution for comparison to assist in refining the dissipation parameter set $\overset{*\alpha}{\beta}_2$. The diaphragm is placed mid-way in a duct of uniform cross-section, with the unit length discretized into $12 \leq M \leq 400$ finite elements R_e^1 of uniform measure Δ. The k=1, M=100 discretization corresponds to the case of Sod [33], with the initial condition specifications $u(x) = 0$, $p = 1 = \rho$ on $0 \leq x \leq 0.5$, $p = 0.1$ and $\rho = 0.125$ on $0.5 < x \leq 1.0$ with $\gamma = 1.4$. Figure 4 graphs the k=1 solution variable set $\{QI(n\Delta t)\}$, at $t = 0.14154s$, as obtained using the order-of-accuracy optimized parameter set $\nu_\alpha^1 \equiv \nu \equiv \nu_\alpha^2, \nu_m^1 \equiv 0, \nu \equiv (15)^{-\frac{1}{2}}$. Each symbol corresponds to a nodal coordinate of q_α^h, and the dashed lines denote the initial conditions. The shock is centered at $x = 0.75$, the contact discontinuity is centered at $x = 0.62$, and the rarefaction wave lies upstream of $x = 0.5$. For comparison, Figure 5 is a graph of the M=99 solution for ρ^h and p^h generated by the Lagrangian-rezone Eulerian "MUSCL" finite difference algorithm of Van Lear [31], compared to the analytical solution shown as solid lines. The k=1 finite element solution is considerably "smoother," with less well defined gradients and plateaus, and with excessive overshoot/undershoot about the shock. Figure 6 is the improved k=1, M=200 finite element algorithm solution, as obtained with the "numerically optimized" dissipation parameter set $\nu_\alpha^1 \equiv \nu\{3/8,0,1/4\}$ and $\nu_\alpha^2 = \nu\{3/4,2,1\}$, for $\nu \equiv (15)^{-\frac{1}{2}}$ and $1 \leq \alpha \leq 3$. Each of the characteristic Riemann solution features is quite accurately represented including accurately planar plateaus and a crisply defined shock with negligible overshoot.

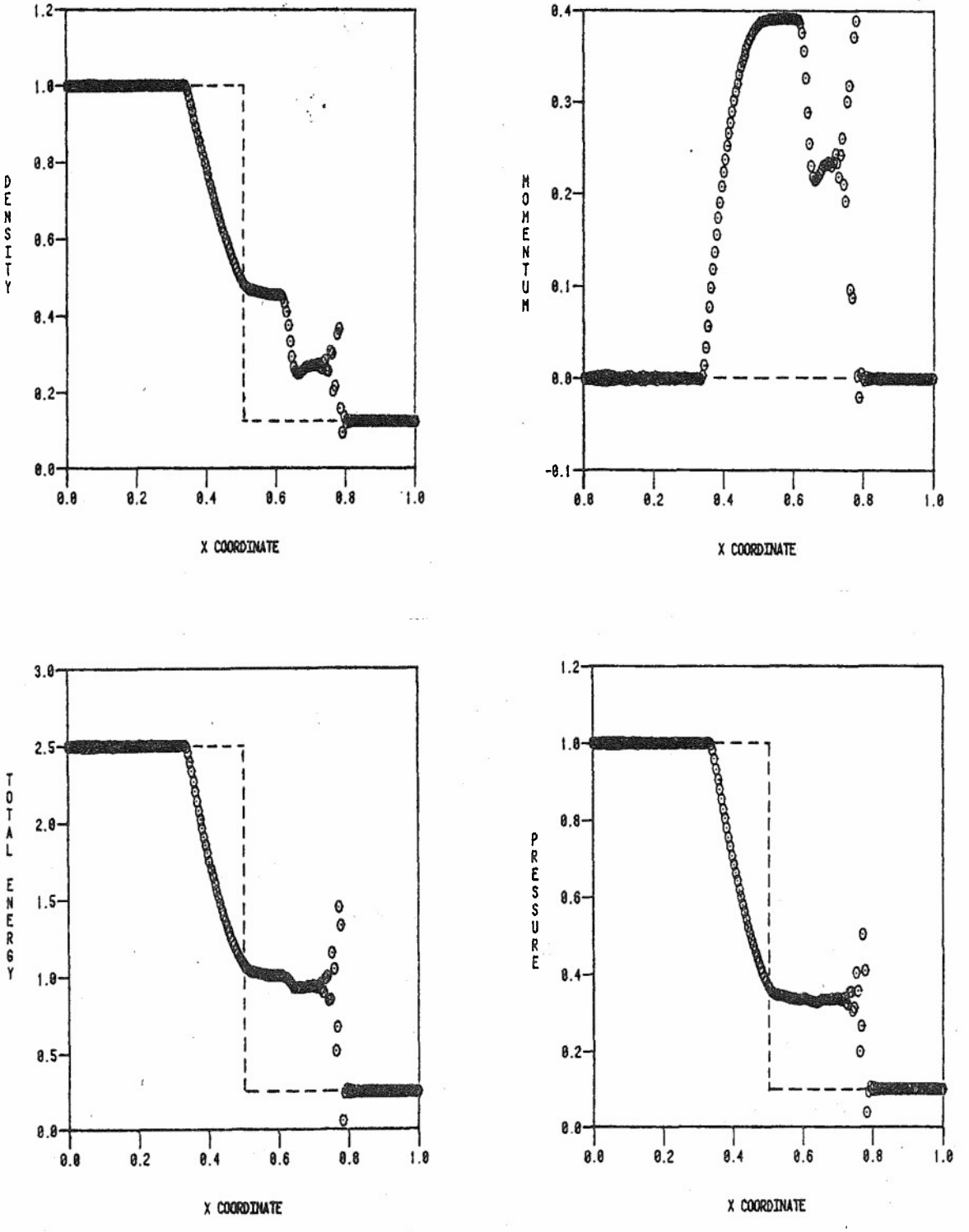


Figure 4. M=200, k=1 Finite Element Algorithm Solution, Riemann Shock Tube, t = 0.14154, $\nu_\alpha^1 \equiv \nu \equiv \nu_\alpha^2$, $\nu_m^1 \equiv 0$, (---) Denotes Initial Conditions.

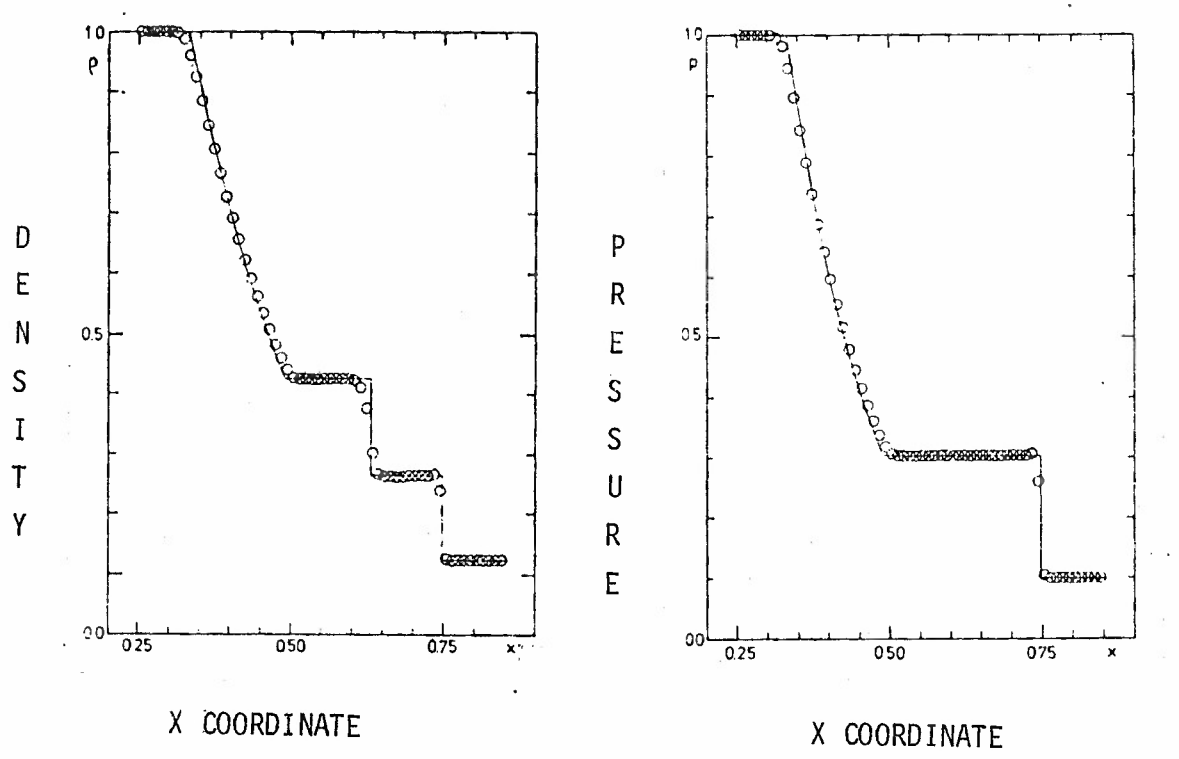


Figure 5. Solution for Riemann Shock Tube Generated By the MUSCL Code, Reported by Van Leer [31], Courant No. = 0.9, $\Delta x = 0.01$, $t = 0.14154s$.

A definitive accuracy assessment can be made in terms of the velocity and internal energy distributions computed from the solution set. Figure 7 compares the M=99 MUSCL code and analytical solution (solid line, Figure 7a), with the $\vec{\nu}_\alpha$ optimized k=1, M=200 and M=100 finite element solutions, Figure 7b)-7c) respectively. In addition, Figure 7d) shows the M=100 data obtained using the Crank-Nicolson finite difference algorithm equivalent of the k=1 finite element algorithm. This familiar finite difference construction as obtained by diagonalizing the [A200] matrices in equation 107-109 and 111-113, which also results in identical elimination of the ν_α^1 terms, cf. [23]. The illustrated solution was generated using $\vec{\nu}_\alpha^2 = \nu\{1,1,1\}$; increasing the dissipation levels would moderate the illustrated excessive shock overshoot, at the expense of farther diffusing the shock over more than the present five domains. In comparison, the M=100, k=1 finite element algorithm results interpolate the shock over only two element domains with negligible overshoot, for the identical CPU and main memory machine requirements. The

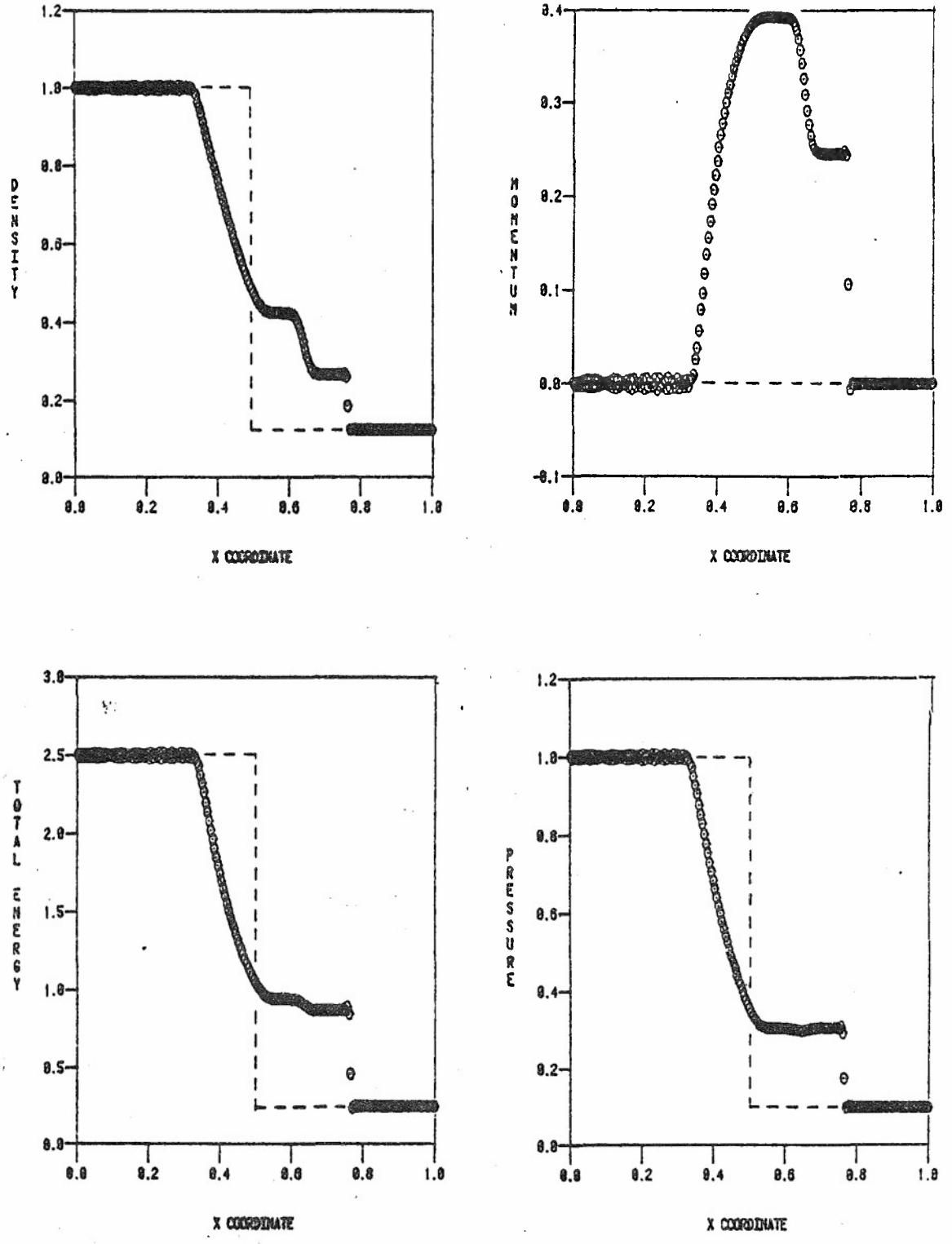
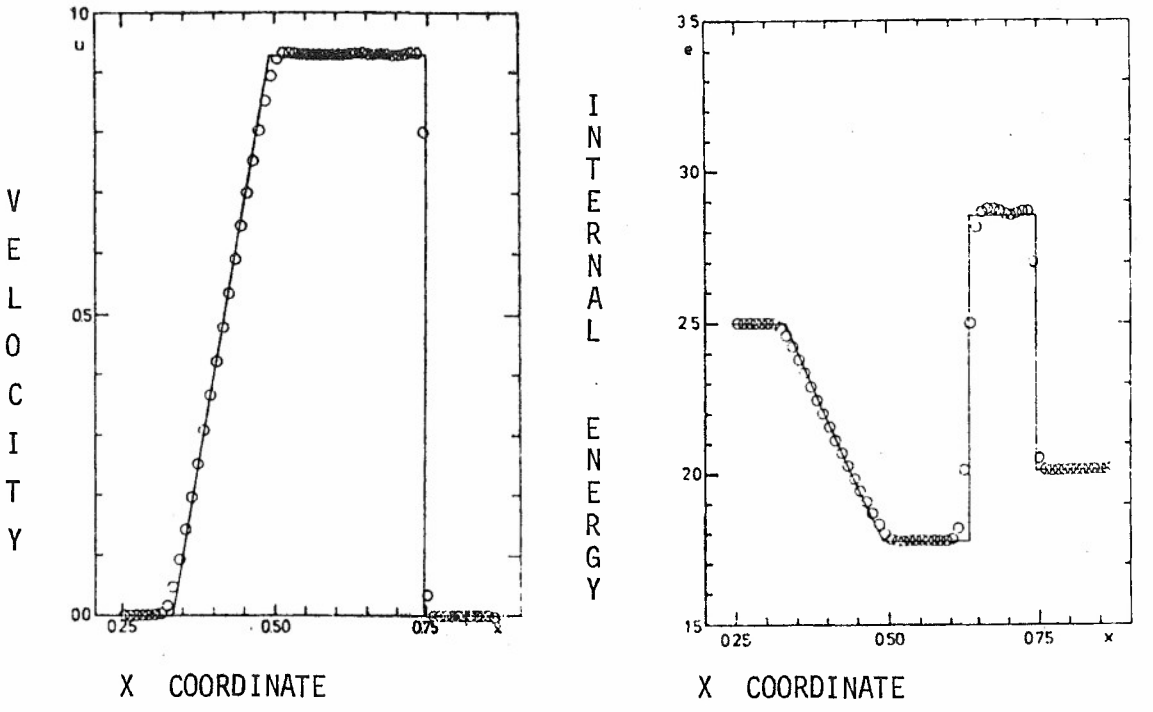
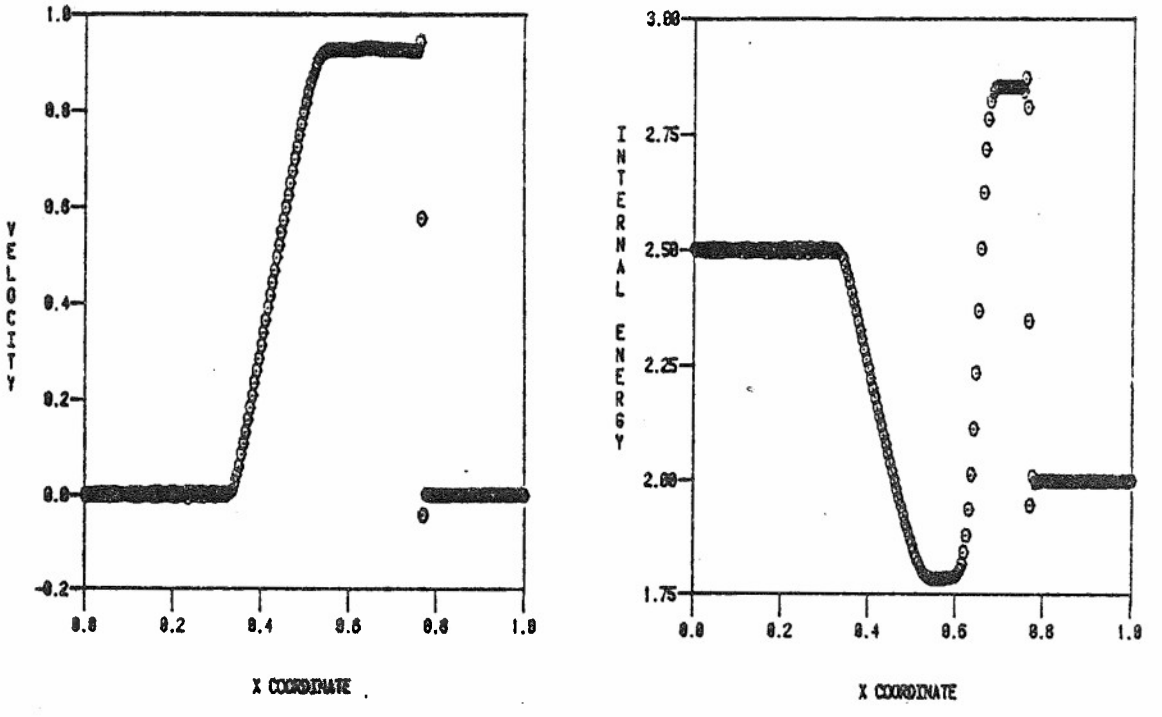


Figure 6. M=200, k=1 Finite Element Solution, Riemann Shock Tube $\nu_\alpha^1 = \nu\{3/8,0,1/4\}$, $\nu_\alpha^2 = \nu\{3/4,2,1\}$, t = 0.14154s (---) Denotes Initial Conditions.

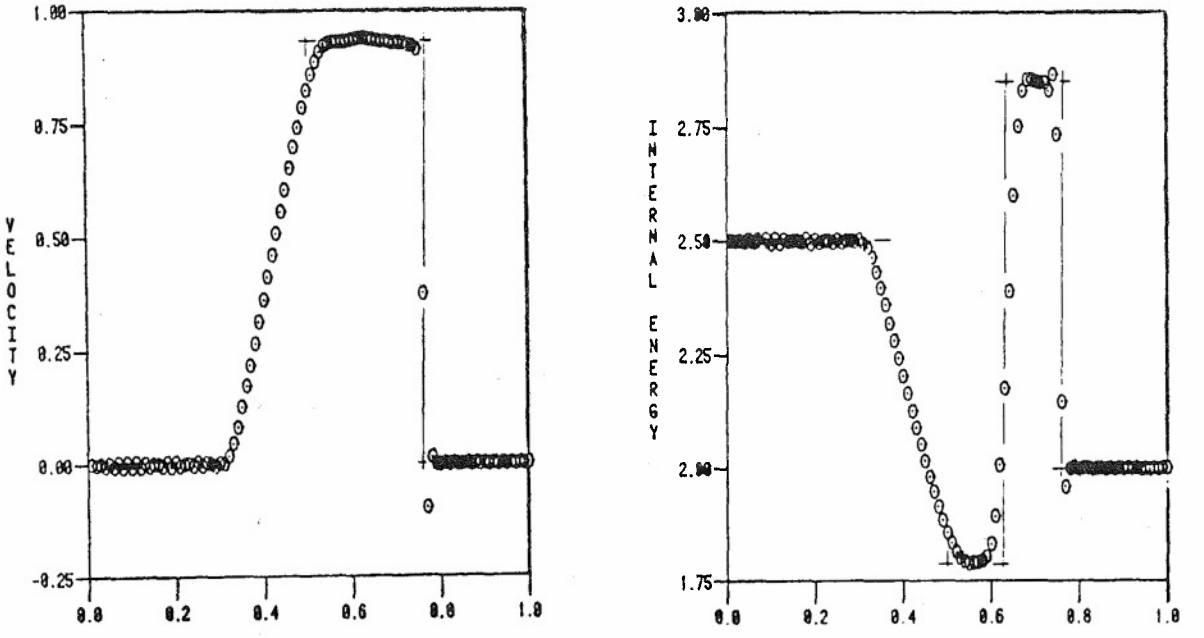


a) MUSCL Code Solution of Van Leer [31], (——) Analytical Solution.

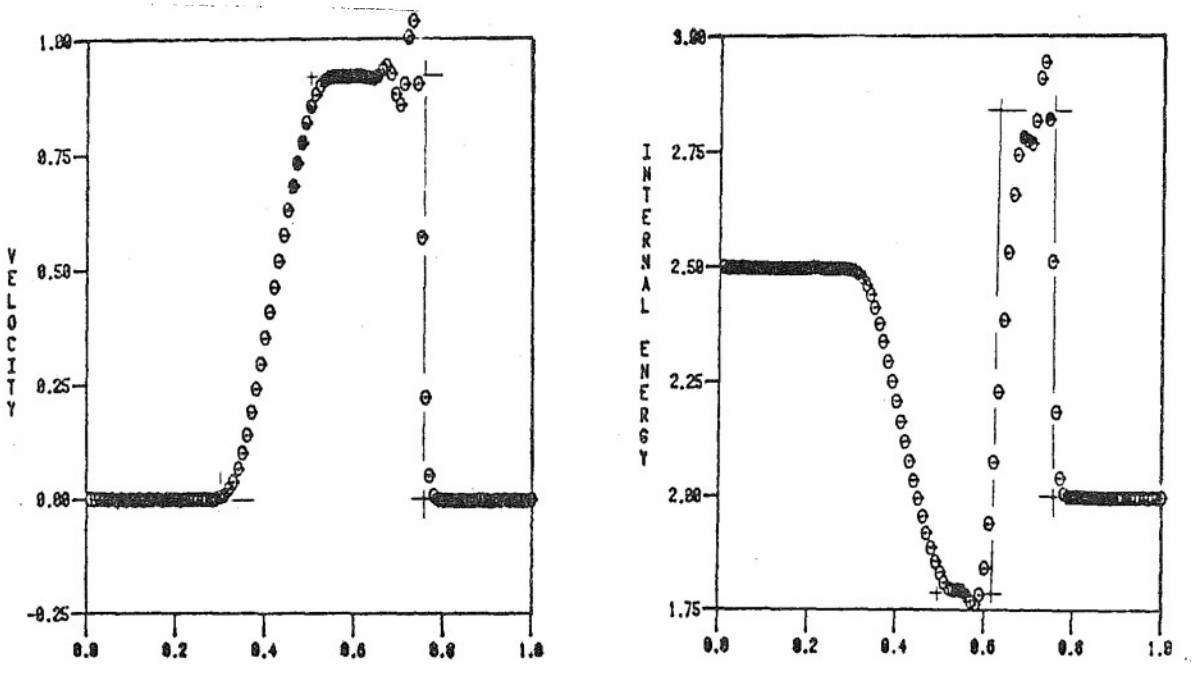


b) M=200, k=1 Finite Element Solution, $\vec{v}_{opt}$.

Figure 7. Finite Element and Finite Difference Algorithm Solution Comparisons, Riemann Shock Tube, t = 0.14154s.



c) M=100, k=1 Finite Element Solution, $\vec{\nu}_{opt}$.



d) Crank-Nicolson Finite Difference Solution, $\vec{\nu}_{\alpha}^{1} = 0.$, $\vec{\nu}_{\alpha}^{2} = (15)^{-\frac{1}{2}}$.

Figure 7. Finite Element and Finite Difference Algorithm Solution Comparisons, Riemann Shock Tube, t = 0.14154s, Concluded.

M=100, k=1 finite element data compare quite favorably to the MUSCL code prediction; the velocity resolution of the shock and the planarity in the high temperature plateau are nominally identical. The MUSCL code data interpolates the contact discontinuity over five domains, while the k=1 data smeared it over nine finite elements. Considering that the Eulerian finite element algorithm is not at all "hard-wired" for this problem, the evidenced accuracy is excellent.

The performance of the k=2 finite element algorithm for the Riemann problem is nominally comparable. Following numerical experimentation, the "optimized" dissipation parameter set was determined as $\nu_\alpha^1 = 0.$ and $\nu_\alpha^2 = \nu\{1/4,3/4,1/2\}$. Figure 8 graphs the resulting solution $\{\underset{\alpha}{QI}(n\Delta t)\}$, at $t = 0.14154s$, as obtained on the M=50 discretization which contains as many nodal coordinates as the k=1, M=100 discretization. Figure 9a graphs the resulting solution parameters velocity and internal energy, and the solid lines are traces of the exact solution. In comparison to the data of Figure 7a)-c), accuracy is excellent, with an improvement in definition of the contact discontinuity evidenced. In the case of the k=2 algorithm, and since $\nu_\alpha^1 = 0$ is "optimal," the finite difference equivalent diagonalized algorithm results are not severely degraded, Figure 9b).

The finite element algorithm Jacobian construction, equations 111-114, exhibits the favorable quadratic convergence rate associated with Newton iteration. For example, Table 2 presents extremum elements of $\{\delta QI\}$ for a typical integration step involving three iterations, and the nodal location of these extrema. Convergence is at least quadratic, including the algebraic pressure equation. The sharply defined solutions, as obtained using minimal levels for ν_α^2, typically require 250 iterations to reach $t = 0.14154s$, at a Courant number ($C = |u+a|\Delta t/\Delta x$) of approximately 0.33.

Since the algorithm is implicit, the sole constraint on Courant number is accuracy. For the discussed results, generated for $C = 0.33$, a Richardson extrapolation step at $t = 0.14154s$ confirmed that the significant digit in $\|q^h\|$ was unaffected by the temporal truncation error associated with C, as required to estimate the semi-discrete (finite element) approximation error. Figure 10 graphs the semi-discrete convergence characteristics, measured in both H^1 and E, of the k=1 and k=2 algorithm solutions for the Riemann problem, over a discretization range.

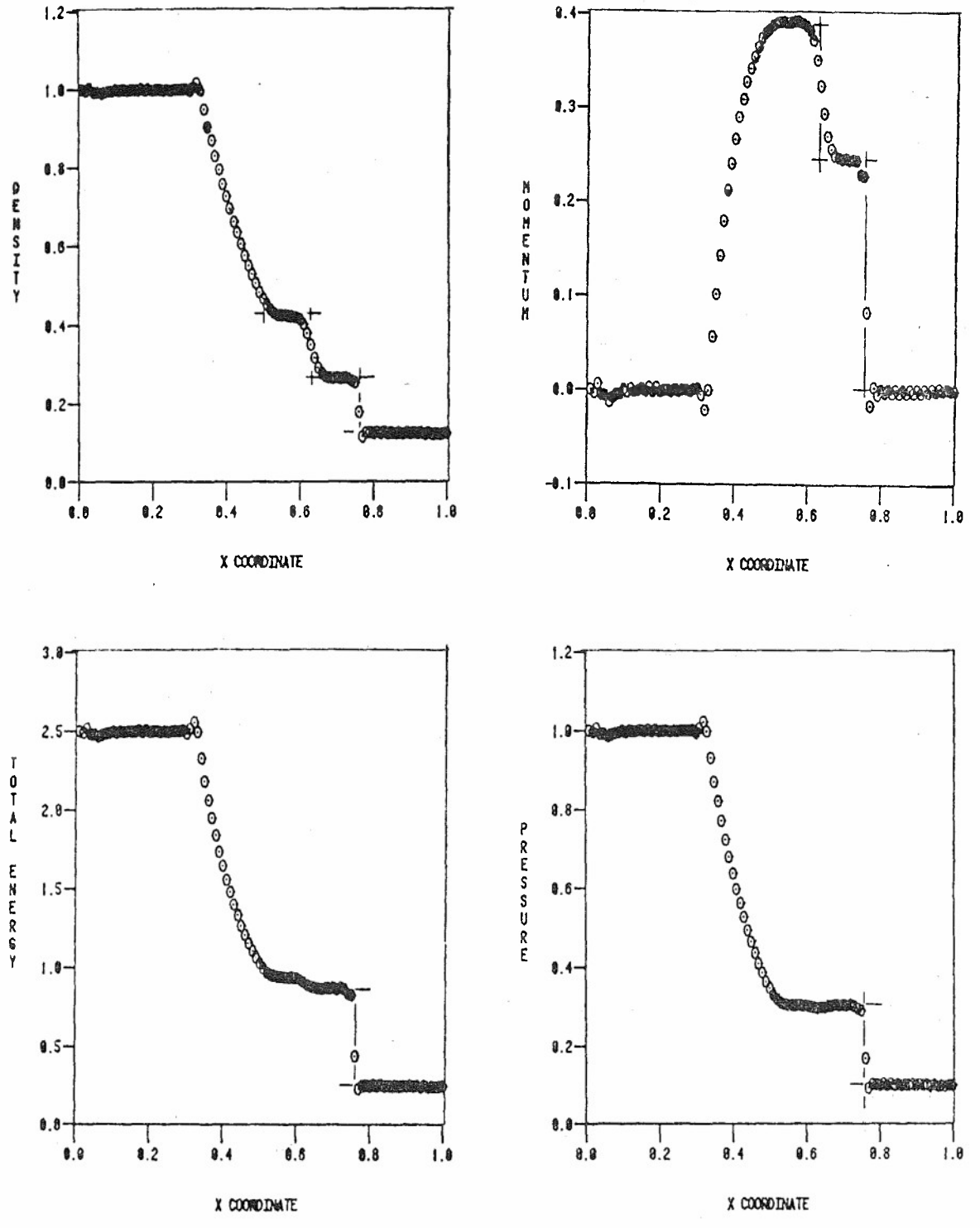
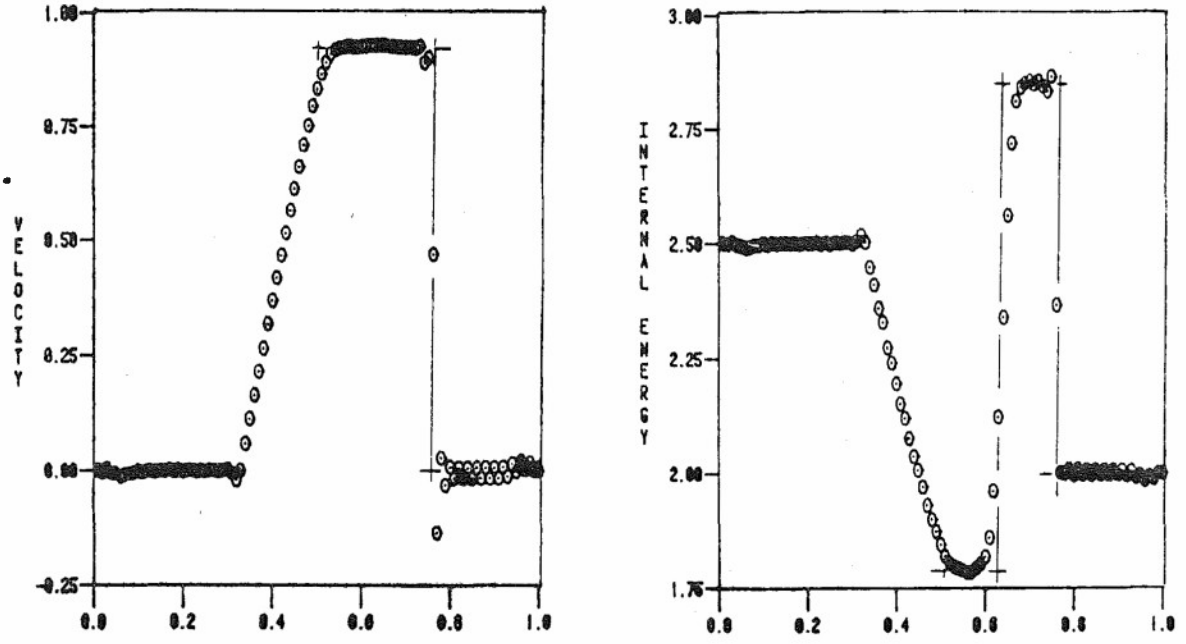
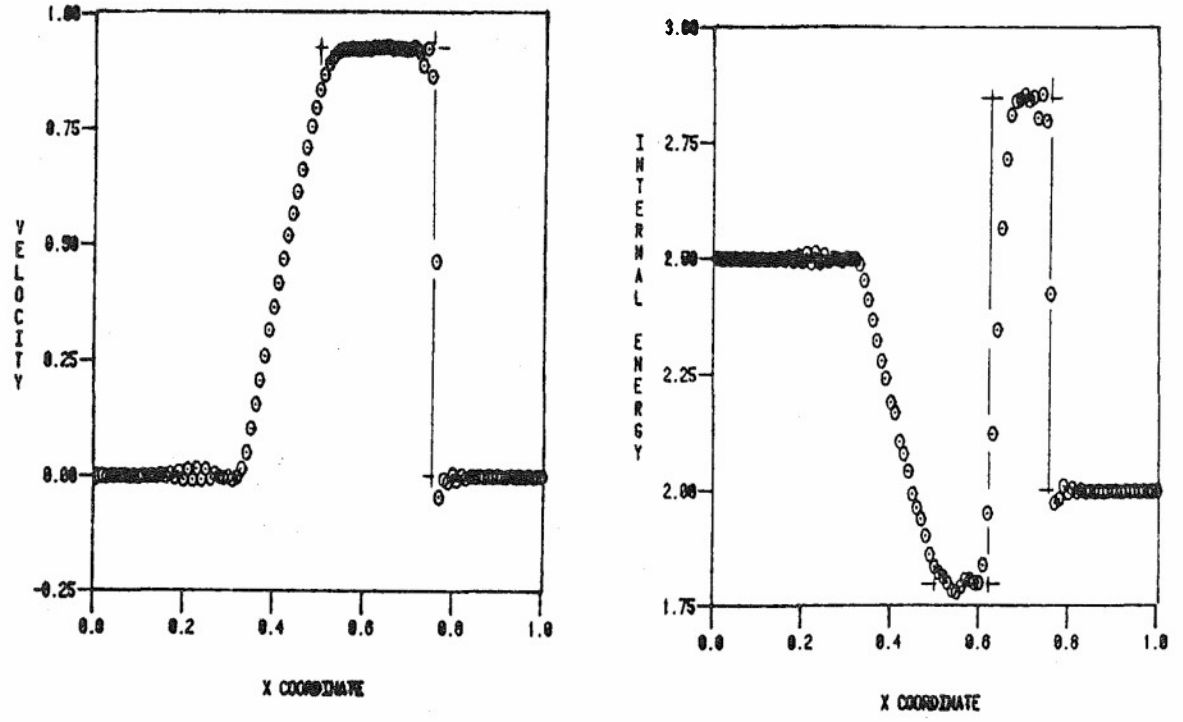


Figure 8. M=50, k=2 Finite Element Algorithm Solution, Riemann Shock Tube, t = 0.14154s, $\nu_{\alpha}^{1} = 0.$, $\nu_{\alpha}^{2} = \nu\{1/4, 3/4, 1/2\}$.



a) Standard Finite Element, $\nu_\alpha^1 = 0.$, $\nu_\alpha^2 = \nu\{1/4, 3/4, 1/2\}$



b) Diagonalized Quadratic, $\nu_\alpha^1 = 0.$, $\nu_\alpha^2 = \nu\{1/4, 3/4, 1/2\}$

Figure 9. Finite Element and Diagonalized Finite Element Algorithm Comparisons, Riemann Shock Tube, M=50, k=2, t = 0.14154s.

TABLE 2

NEWTON ITERATION CONVERGENCE IN {δQI}

RIEMANN SHOCK TUBE

M=100, k=1, ε = 0.001

Iteration	{δR}max	Node	{δM}max	Node
1	-0.87325E-02	70	-0.20780E-01	70
2	0.13072E-02	69	0.31246E-02	69
3	0.21915E-03	69	0.69326E-03	69

Iteration	{δG}max	Node	{δP}max	Node
1	-0.23351E-01	70	-0.75154E-01	70
2	0.34658E-02	70	-0.51826E-02	71
3	0.46731E-03	70	-0.50114E-03	68

The dashed lines, interpolating the k=2 data, lie uniformly above the k=1 data (solid lines), for each dependent variable, and each has a nominally identical slope. A modest slope distribution exists among the various dependent variable data. Assuming this distinction insignificant, a nominal convergence rate of 2/3 is indicated, in qualitative agreement with a difference scheme of p^{th} order accuracy, ie.,

$$\|q_\alpha^h\| \leq C_1 \Delta^{p/p+1} + \ldots \tag{115}$$

where p = 2 and C_1 is a constant independent of Δ, the measure of the uniform mesh. The three solid symbols on $\Delta_e = 10^{-2}$ are the data in E generated by the M=100 finite difference solution shown in Figure 7d). The consistent

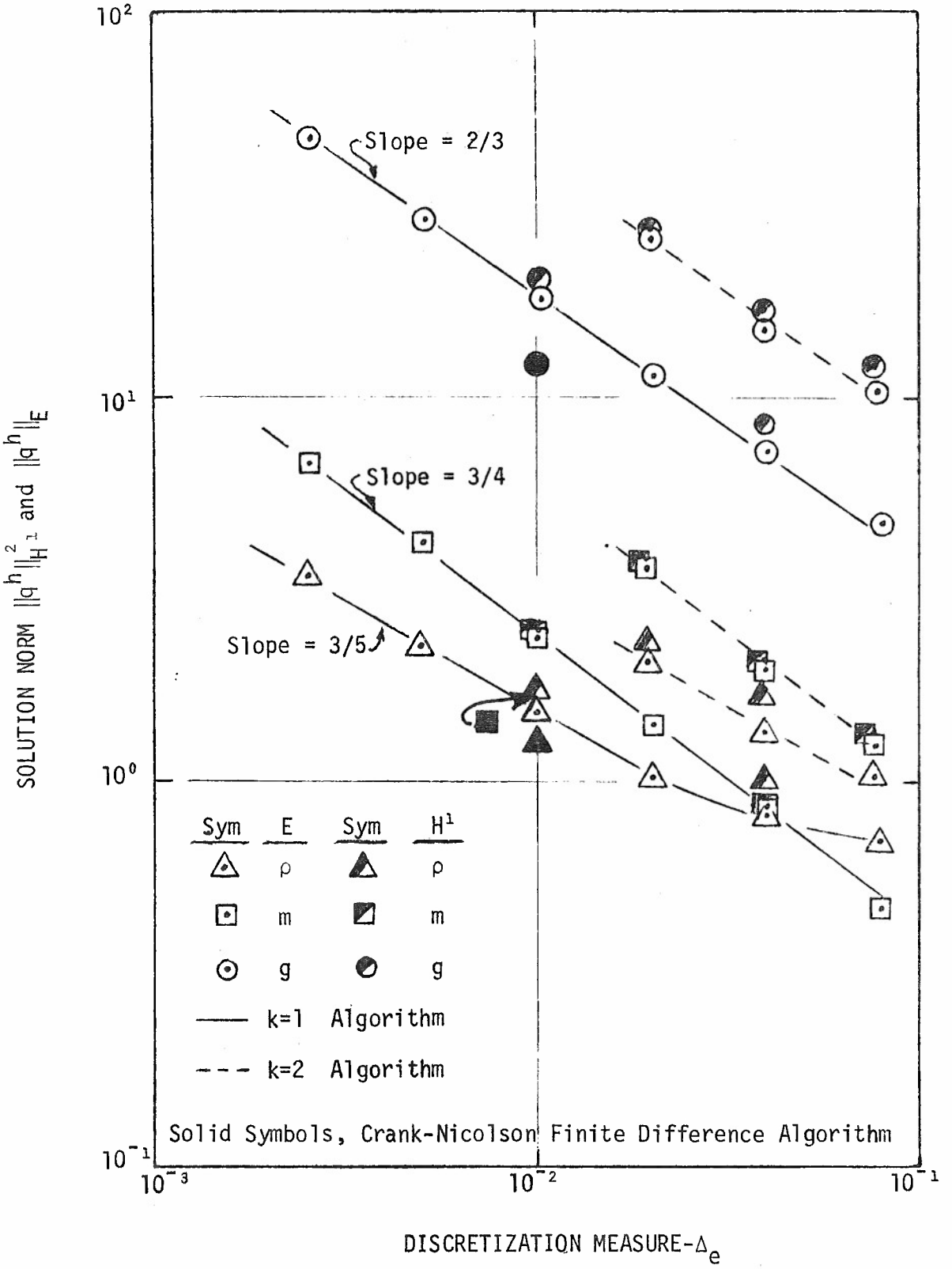


Figure 10. Semi-Discrete Approximation Accuracy And Convergence in $\|q^h\|^2_{H^1}$ and $\|q^h\|_E$ Finite Element Algorithm Solution For Riemann Shock Tube.

trend in error extremization by a finite element algorithm thus appears confirmed for this problem (class).

Solution efficiency is principally determined by discretization (M) and integration timestep (C), in concert with the requested convergence level for the Newton iteration algorithm. Table 3 summarizes the effect of decreasing solution cost by increasing C for the M=100, k=1 Riemann solution. Solution cost can be nominally halved by increasing the Courant number to a certain point. Thereafter, diminishing returns are encountered as additional Newton iterations are required for convergence. In all data for C > 0.33, the significant digit in the semi-discrete approximation norm is being directly affected by temporal truncation error, and solution accuracy is degraded monotonically with increasing C. Of particular note, the k=1 finite element algorithm solution accuracy at C = 1.41 is uniformly superior to the C=0.33 finite difference solution, see Figure 10, and was obtained at 40% of the computer cost.

TABLE 3

ACCURACY AND EFFICIENCY SUMMARY
RIEMANN SHOCK TUBE
M=100, k=1, ε=0.001

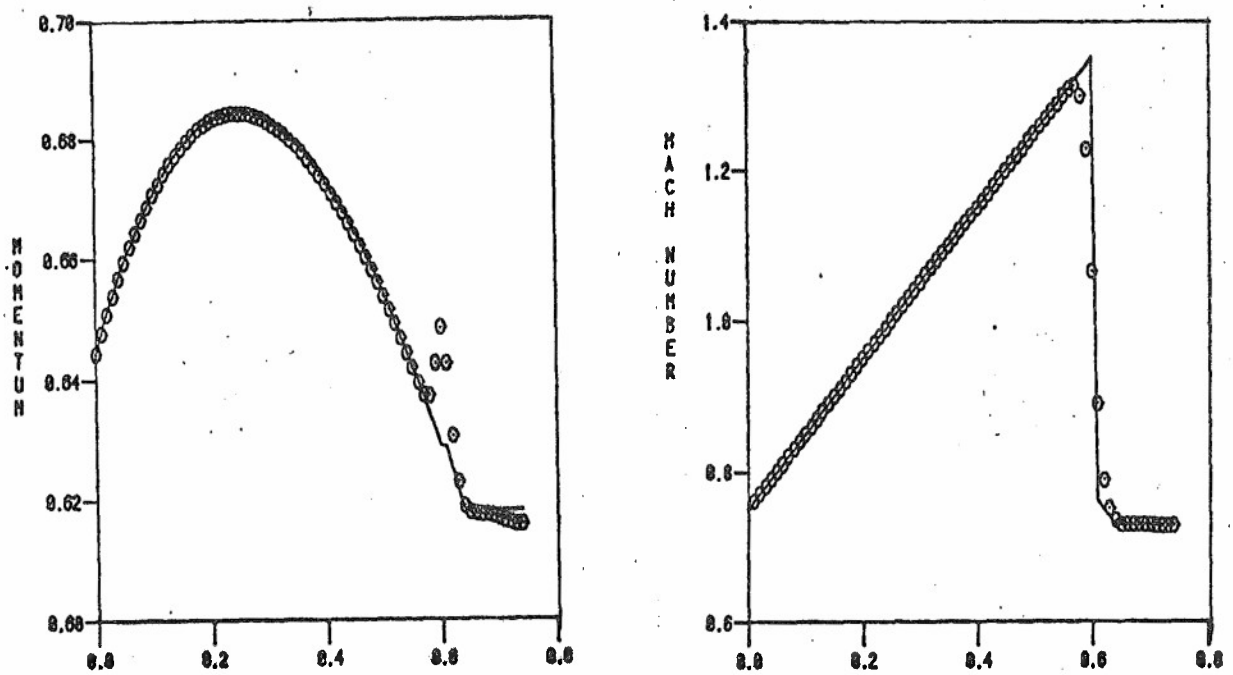
Courant Number	No. Time Steps	No. of Algorithm Passes	GO Step CPU (Normalized)	Energy Norm $\|q_\alpha^h\|_E$		
				ρ	m	g
0.33	80	229	1.00	1.54	2.43	17.9
0.44	60	180	0.79	1.50	2.39	17.4
0.67	40	121	0.54	1.46	2.29	16.8
0.90	30	92	0.42	1.41	2.15	16.0
1.41	20	82	0.38	1.34	1.90	14.4

3. Mixed Flow In A Variable Cross-Section Duct

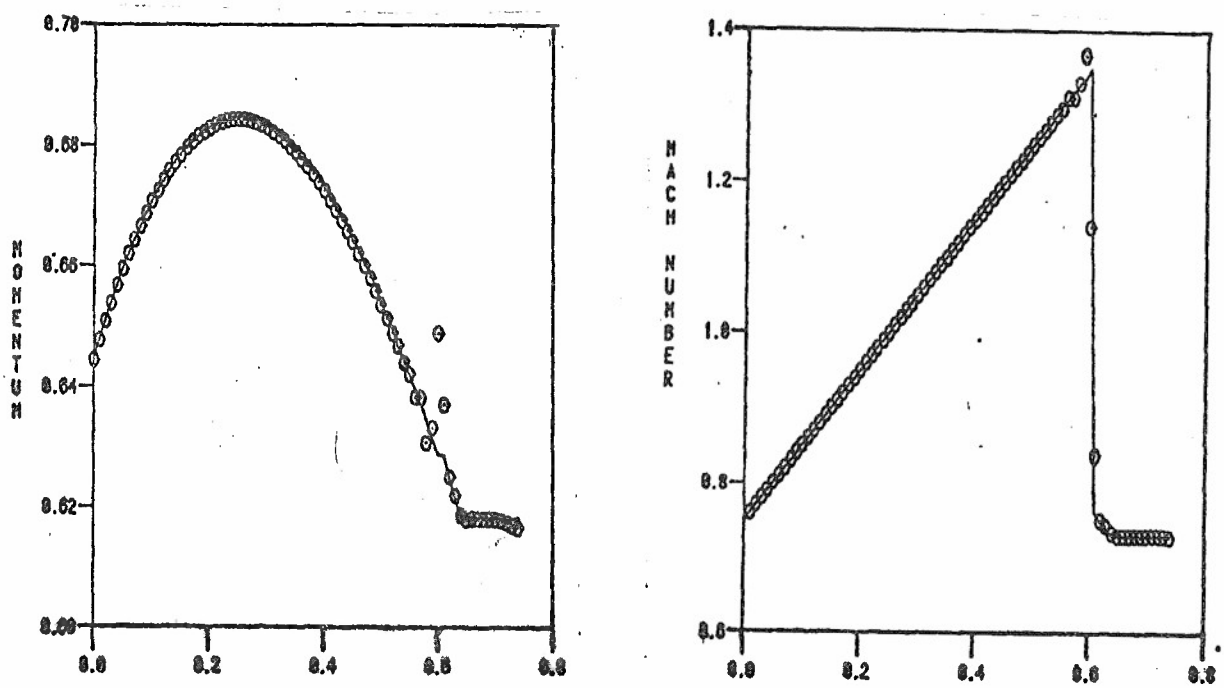
Since the finite element algorithm is implicit, all solution zones are in continuous communication, even those regions where the flow is locally supersonic. An interesting test occurs for a mixed flow situation, wherein subsonic boundary data modification require an adjustment to the flow in a supersonic region. As an example, off-design operation of a de Laval nozzle yields a strong shock in the divergent section, such that the shocked flow can diffuse to the exit chamber pressure. A modest increase in the chamber pressure requires the shock to relocate upstream of its original location, wherein the flow was supersonic, with the shock always separating the mixed flow regions. Anytime during this period of adjustment, the supersonic flow upstream of the shock must not respond to the new chamber pressure level, the shock must progressively weaken as it moves upstream through regions of smaller cross-section, and the downstream flow must diffuse smoothly to the new exit conditions.

Figure 11 summarizes the M=74, k=1 algorithm steady-state solution, for an off-design nozzle flow with $M_1 = 1.35$ upstream of the shock, using $\vec{\nu}_{opt}$ and $\frac{1}{2}\vec{\nu}_{opt}$. The solid line is the exact solution; the $\frac{1}{2}\vec{\nu}_{opt}$ solution is in excellent agreement on shock strength and definition. It is interesting how the momentum solution attempts to interpolate the discontinuity of the shock, which of course will be of finite width. The subsonic inlet boundary conditions are specified ρ, m, and g with p computed from equation 7. The subsonic outlet condition is p specified, and a vanishing normal derivative for ρ, m, and g.

Using the solution of Figure 11 as initial conditions, the specified exit pressure was raised 15% and held fixed at the new level. The inlet conditions remained as originally specified. Figure 12 graphs the k=1 algorithm solution at the new steady-state, where the solid lines are the initial conditions. The flow has adjusted fully to the new exit pressure, with a new shock Mach number of $M_1 = 1.15$. The flow upstream of the shock is unaltered from the initial conditions, while a smooth subsonic diffusion to exit conditions is predicted downstream. The shock is interpolated across four elements with no overshoot. Figure 13 summarizes the comparison M=37, k=2 algorithm solution, as obtained using $\vec{\nu}_\alpha = \frac{1}{2}\vec{\nu}_{opt}$. The solution accuracies are indistinguishable.



a) M=74, k=1, $\vec{\nu}_\alpha = \vec{\nu}_{opt}$



b) M=74, k=1, $\vec{\nu}_\alpha = \frac{1}{2}\vec{\nu}_{opt}$

Figure 11. Finite Element Algorithm Steady-State Solution, Off-Design Nozzle Flow.

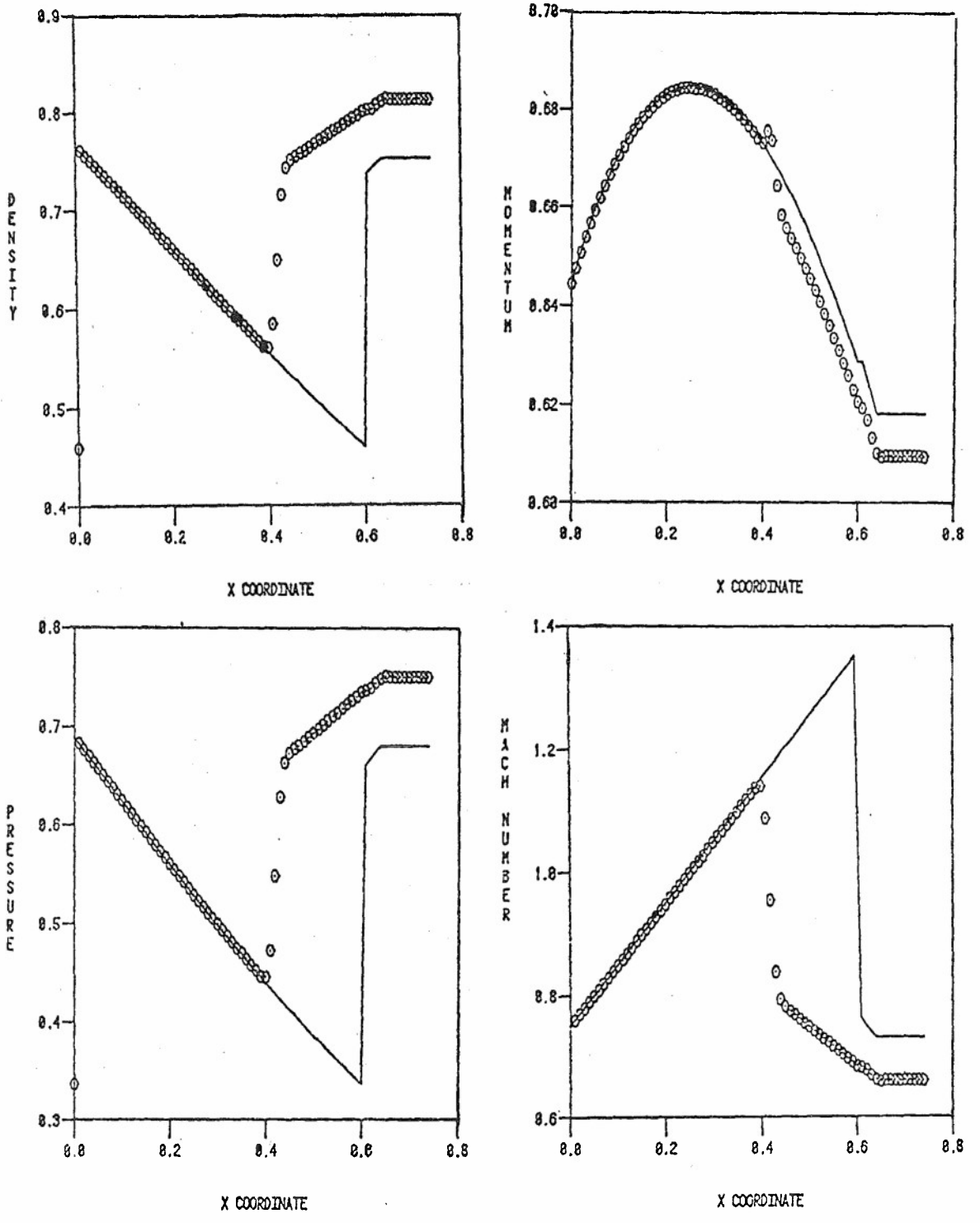


Figure 12. M=74, k=1 Algorithm Solution for Mixed Flow in a Variable Cross-Section Duct, $\vec{\nu}_\alpha = \frac{1}{2}\vec{\nu}_{opt}$, (——) Initial Condition.

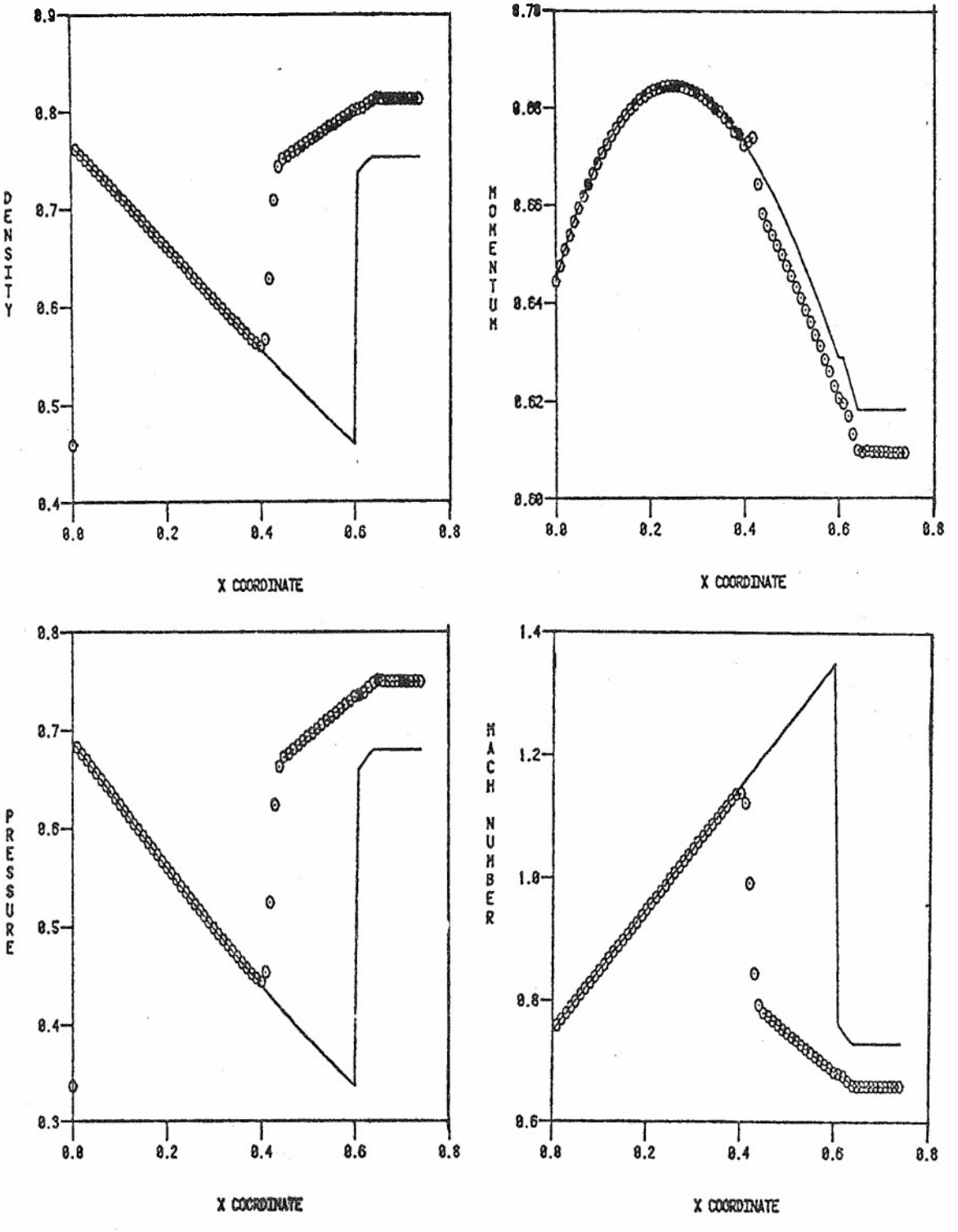


Figure 13. M=37, k=2 Algorithm Solution for Mixed Flow in a Variable Cross-Section Duct, $\vec{\nu}_\alpha = \frac{1}{2}\vec{\nu}_{opt}$, (——) Initial Condition.

4. Two-Dimensional Flow, Metric Data

As noted in the Introduction, a wide variety of methodologies exist for generation of the inverse coordinate transformation $\eta_i = \eta_i(x_j)$, see equation 28. For example, Figure 14 illustrates body-fitted coordinate systems for various two-dimensional aerodynamic configurations, as generated using Poisson equation solutions [8]. The principal requirement is to make contours corresponding to aerodynamic surfaces, and/or symmetry or periodocity boundaries, be coordinate curves of the η_i system. The second requirement, from the standpoint of simulation accuracy, is to avoid highly distorted computational cells in critical flow regions, eg., stagnation points.

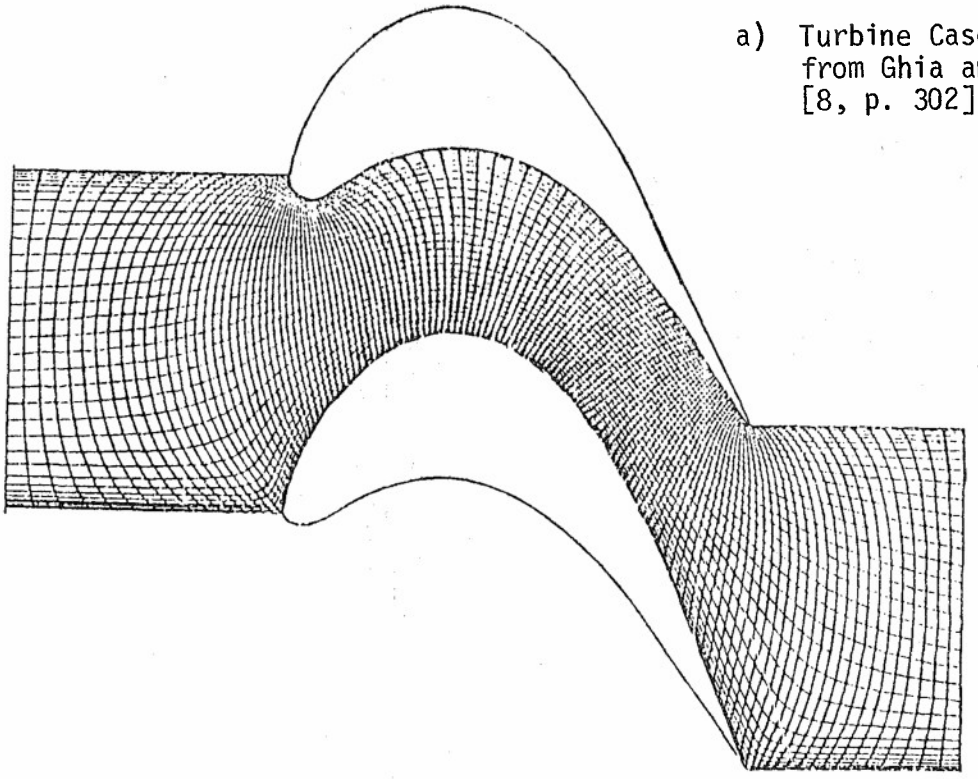
With respect to the generalized coordinates finite element algorithm, each of the grids illustrated in Figure 14 is the definition of the array of nodal coordinates $\{XI\} \equiv \underset{e}{\Sigma}\{XI\}_e$, $1 \le e \le M$, $1 \le I \le 2$, required to define the coordinate transformation, see equations 29-32. Restricting attention to the bilinear basis $\{N_1(\vec{\eta})\}$, k=1 in equation 29, the typical finite element domain R_e^2 contains only vertex nodes. For the counterclockwise numbering convention, Figure 1, denote the element node coordinates as $\{XI\}_e \equiv \{XI, YI, 1 \le I \le 4\}$. The array $\{ETAKJ\}_e$, $1 \le (K,J) \le 2$, is constructed via equation 34, and the calculus operation is elementary. Define the discrete approximation on R_e^2 using $\{N_1(\eta)\}$ as

$$\left\{\frac{\partial \eta_k}{\partial x_j}\right\}_e \equiv \frac{1}{\det J_e}\{N_1\}^T\{ETAKJ\}_e, \qquad 1 \le (K,J) \le 2 \tag{116}$$

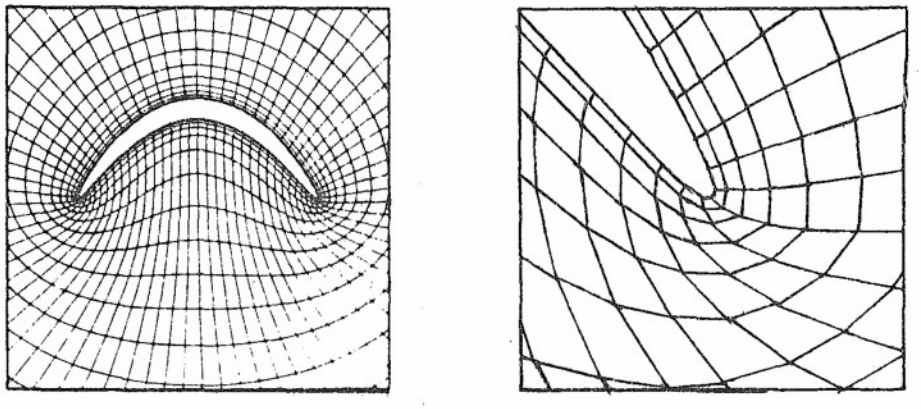
Noting that $(\det J)_e^{-1}$ cancels in the algebra, the elements of $\{ETAKJ\}_e$ are

$$\{ETAKJ\}_e^T = \frac{1}{2}\begin{Bmatrix} \{Y4\text{-}Y1,\ Y3\text{-}Y2,\ Y3\text{-}Y2,\ Y4\text{-}Y1\}_e,\ (1,1) \\ \{X1\text{-}X4,\ X2\text{-}X3,\ X2\text{-}X3,\ X1\text{-}X4\}_e,\ (1,2) \\ \{Y1\text{-}Y2,\ Y1\text{-}Y2,\ Y4\text{-}Y3,\ Y4\text{-}Y3\}_e,\ (2,1) \\ \{X2\text{-}X1,\ X2\text{-}X1,\ X3\text{-}X4,\ X3\text{-}X4\}_e,\ (2,2) \end{Bmatrix} \tag{117}$$

where the numbers in parenthesis indicate (K,J).

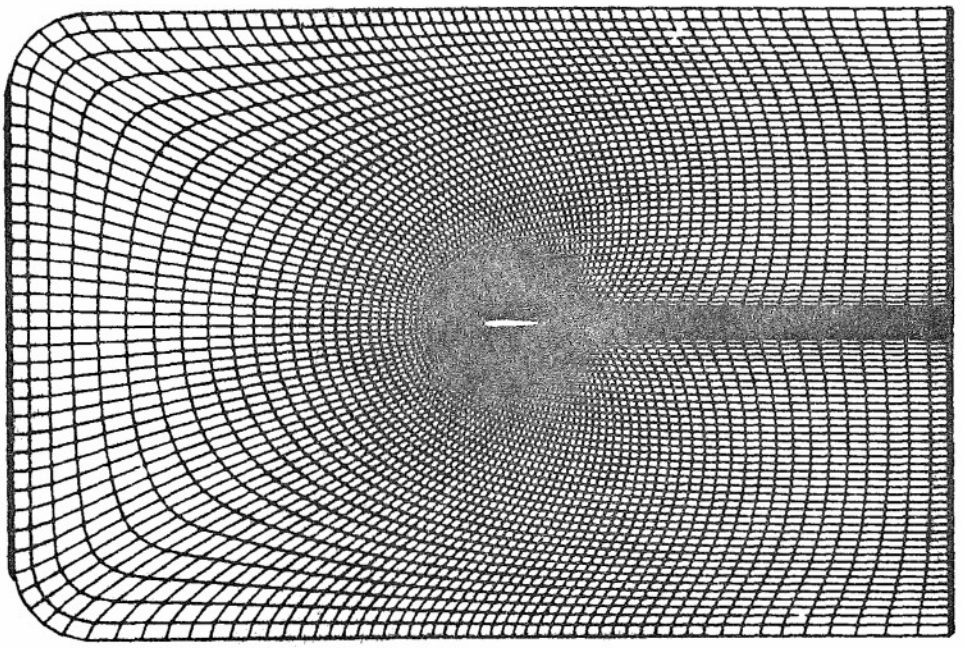


a) Turbine Cascade, from Ghia and Ghia, [8, p. 302]

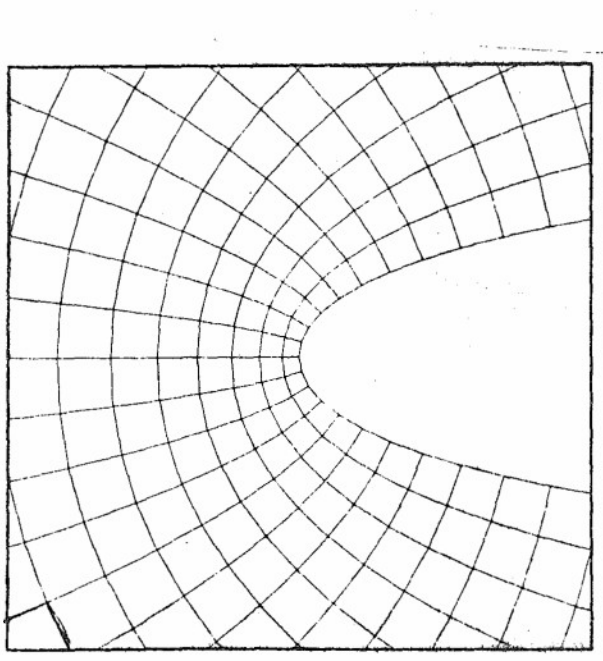


b) Highly Cambered Airfoil and Trailing Edge Close-Up, from Sorenson and Steger [8, p. 456]

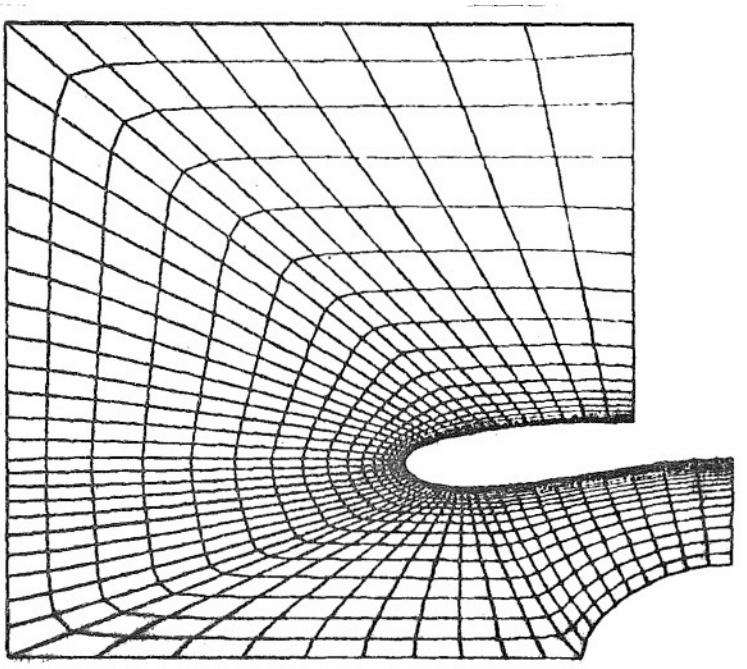
Figure 14. Examples of Computational Grid Transformations For Two-Dimensional Aerodynamic Flow Predictions.



c) C-Type Grid For Airfoil in a Wind Tunnel, from Sorenson and Steger [8, p. 458]



d) Leading Edge Detail Sorenson and Steger [8, p. 457]



e) C-Type Grid For Nacelle With Centerbody, from Kowalski [8, p. 349].

Figure 14. Examples of Computational Grid Transformation For Two-Dimensional Aerodynamic Flow Prediction, Concluded.

The corresponding definition for $\det[J]_e$ is

$$\det[J]_e \equiv \{N_1\}^T \{DET\}_e \tag{118}$$

and

$$\{DET\}_e = \frac{1}{4} \begin{Bmatrix} (X2-X1)(Y4-Y1) - (X4-X1)(Y2-Y1) \\ (X2-X1)(Y3-Y2) - (X3-X2)(Y2-Y1) \\ (X3-X4)(Y3-Y2) - (X3-X2)(Y3-Y4) \\ (X3-X4)(Y4-Y1) - (X4-X1)(Y3-Y4) \end{Bmatrix}_e \tag{119}$$

The contravariant convection velocity approximation $\bar{u}_k^h$, equation 39, on R_e^2 is,

$$\bar{u}_k^e \equiv \{N_1\}^T \{UBARK\}_e \tag{120}$$

and the algebra operations yield.

$$\{UBARK\}_e = \{ETAKJ\}_e \{N_1\}^T \{UJ\}_e \tag{121}$$

Equation 121 is evaluated on a nodal basis, whereupon $\{N_1\}$ reduces to the Kronecker delta. Summation is implied over J, and at the nodes UJ $\equiv$ MJ/R, ie., $u_i \equiv m_i/\rho$.

The tensor product Jacobian formulation, given for the quasi-one-dimensional flow, is modified only to account for alignment with either the η_1 or η_2 coordinate axis. The cardinal basis remains as given in equation 104, and DET_e is equal to one-half the element measure, see equation 106. Hence, for equations 66-71, and using Pythagorus' rule

$$\{DETK\}_e = \tfrac{1}{2}\sqrt{(XKR - XKL)^2 + (YKR - YKL)^2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \tag{122}$$

where R and L denote "right" and "left" respectively, for η_k alignment. Using the same considerations, the elements of $\{ETAKJ\}_e$ are equivalent to the direction cosines of the local affine transformation between x_i and η_j. Denoting Θ as

the angle between the x_1 and η_1 axes on R_e^1, for example, then

$$\{ETAKJ\}_e^T = \begin{Bmatrix} \{\cos\theta, \sin\theta\}, & (1,1) \\ \{-\sin\theta, \cos\theta\}, & (1,2) \end{Bmatrix} \tag{123}$$

Using equation 123, the elemental definition for convection velocity in $[J_\eta]$ remains

$$\{UBARK\}_e = \{ETAKJ\}_e\{N_1\}^T\{UJ\}_e \tag{124}$$

5. Phase Accuracy, The Rotating Cone

The rotating cone test case is a multi-dimensional solution of the linear hyperbolic continuity equation 1. A given initial density distribution $\rho^0(\vec{x},0)$ is subject to an imposed constant convection velocity field $\vec{U}_0 \equiv r\omega\hat{\theta}$, where ω is the angular velocity. The analytical solution is exact preservation of ρ^0 for all time, and the test is extremely demanding with respect to algorithm phase accuracy and embedded artificial diffusion. The initial distribution will be rapidly destroyed by either error mechanism.

The finite element tenson product algorithm, with zero artificial dissipation, $\vec{\beta}_2 \equiv \vec{0}$ in equation 22, can produce solutions to the rotating cone test problem exhibiting excellent accuracy. The solution domain R^2 is the unit square, discretized uniformly into an M = 32 x 32 mesh, for the k=1 algorithm, and an M=16x16 mesh for the k=2 algorithm. Both meshes thus possess 33x33 = 1089 nodal coordinates. The initial density distribution is a "cosine hill," obtained by rotating a cosine distribution about the peak level. Figure 15a illustrates the resultant distribution for $\rho^0(x,y)$, as obtained for interpolating the cosine half-wave onto seven nodal coordinates (with levels 15, 50, 85, 100, 85, 50, 15). A non-reflecting, vanishing normal derivative boundary condition is employed, $a_1^1 \equiv 0 = a_3^1$ in equation 18.

Figure 15 summarizes the various numerical solutions, as a function of Courant number C, and semi-discrete approximation function degree k, following 2π rotation from the initial condition specification. For comparison,

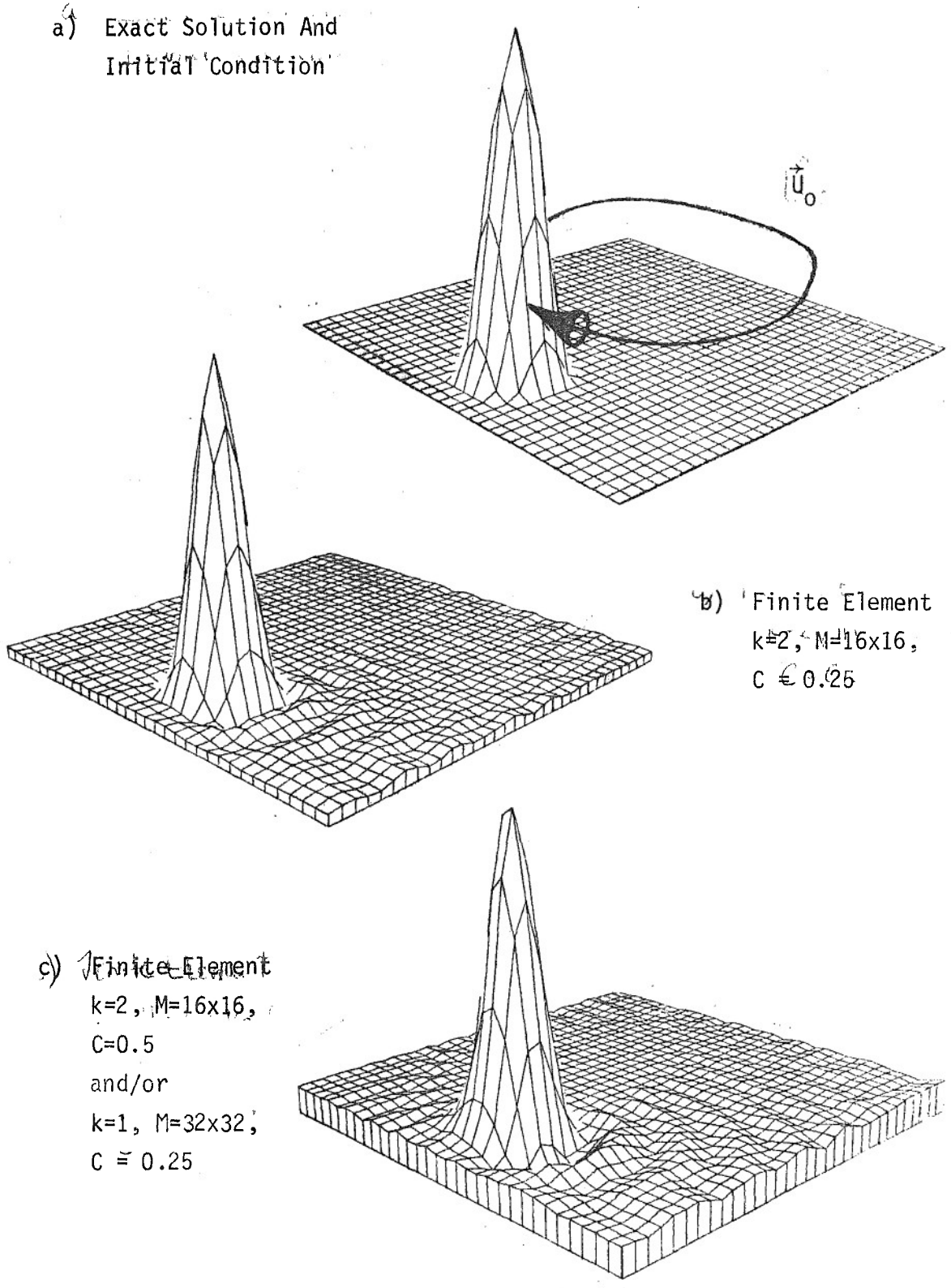
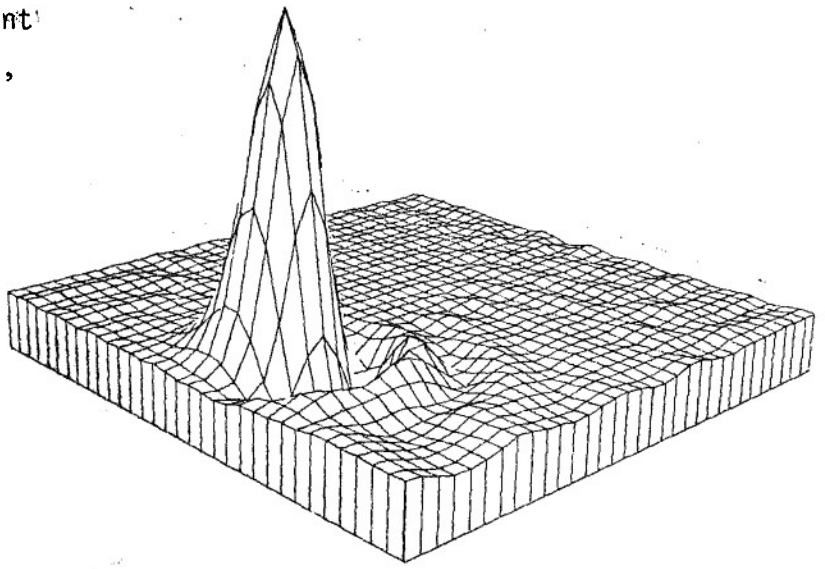
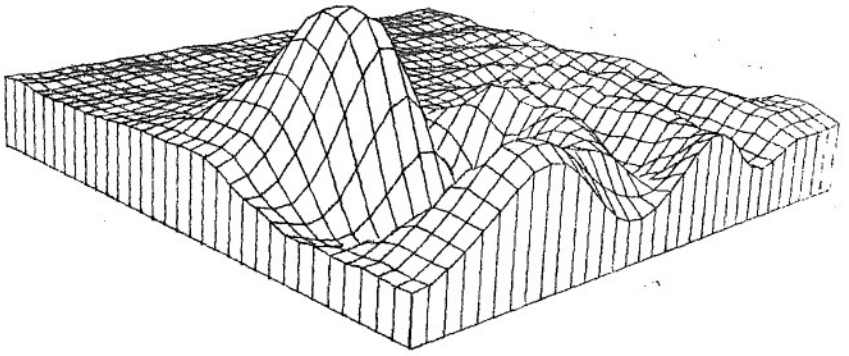


Figure 15. Rotational Convection of a Cosine Hill Distribution.

d) Finite Element
k=1, M=32x32,
C = 0.5





e) Finite Difference
Crank-Nicolson
(K=1) Equivalent
M=32x32, C = 0.25

f) Finite Difference
k=2 Equivalent
M=16x16,
C = 0.5

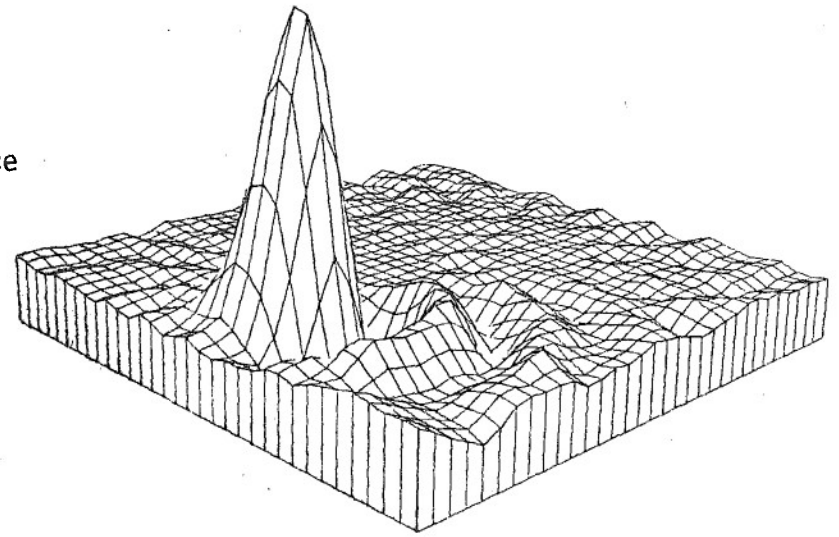


Figure 15. Rotational Convection of a Cosine Hill Distribution (Concluded).

Figure 15a) is also the exact analytical solution, and shows the sense of rotation induced by $\vec{U}_0$. Figure 15b) is the solution generated by the non-dissipative k=2 algorithm, using a constant integration time step Δt yielding C = 0.25 at the center of $\rho(\vec{x},t)$. (Note that C varies linearly, from zero at the center of R^2, as a consequence of $\vec{U}_0$.) The accuracy of this solution is excellent; the final distribution exhibits almost exactly the symmetries of ρ^0, the peak level has actually increased slightly to 105, and the extremum magnitude of the trailing phase dispersion error wake is 2% and quantized by the height of the plot base of R^2. The modest background waviness, on the plane exterior to ρ^0, is the residual dispersion induced by the passage of the solution.

Increasing the integration time-step $\Delta t(C)$, and/or decreasing the semi-discrete approximation degree k, degrades the accuracy of the solution. For example, Figure 15c) graphs the final solution $\rho(\vec{x},t_f)$, as obtained by the k=2 algorithm at C = 0.5, and/or the k=1 algorithm at C = 0.25. In both cases, the additional phase distortion has retarded the solution group velocity, such that the peak has not reached the correct nodal coordinate by t_f, hence the flattened appearance. However, marching forward an additional one or two Δt recovers the solution extremum of 100, for k=2, and 98 for k=1, at a nodal coordinate. The maximum amplitude of the trailing dispersion wake is 9%, note the higher plot base, and the background ripple structure throughout R^2 is modestly more pronounced. Figure 15d) graphs the k=1 algorithm solution obtained for C = 0.5. The indicated solution peak is 93, and is located one nodal coordinate shy of where it should be. The extremum dispersion wake error is 17%, hence the rather pronounced plot base. However, the solution accuracy is still quite acceptable.

For comparison, Figure 15e) graphs the solution obtained for C = 0.25 by the Crank-Nicolson finite difference equivalent of the non-dissipative k=1 algorithm. This algorithm is obtained simply by diagonalizing the matrix [B200], equation 74. This ad hoc operation destroys the phase accuracy, recall Figure 3, and hence yields the extremely poor solution for the rotating cone. The peak has decayed to 47, the extremum wake error magnitude is 25% of ρ^0, and the phase error is well dispersed over R^2. For completeness, Figure 15f)

graphs the solution for the Crank-Nicolson equivalent of the k=2 algorithm at C = 0.5. The peak is 85, the wake extremum is 21%, and the dispersion error is well dispersed onto R^2.

These computational results confirm the phase accuracy robustness of the developed finite element algorithm construction. This attribute is deemed particularly important when addressing high Reynolds number flow prediction. The results of several additional tests are reported [15], wherein various combinations of ν_1^1 and ν_1^2 were employed for k=1 solutions of the rotating cone test. The magnitude of the extremum wake error could be reduced, using the dissipation mechanism, but always at the expense of diffusing the solution peak as well. From the standpoint of accuracy, these data strongly confirm that poor algorithm phase accuracy cannot be corrected through use of artificial dissipation.

6. Two-Dimensional Riemann Shock Tube.

Several additional aspects regarding algorithm accuracy and performance aspects can be assessed using as a test case the two-dimensional equivalent of the Riemann shock tube. Using the convergence data discussed in Section V.2, a reasonable (affordable) discretization of the Riemann problem was defined as M=32xN, where $4 \leq N \leq 20$ dependent upon whether the test problem was inviscid or viscous. For reference, Figure 16 graphs the one-dimensional, k=1 algorithm solution parameters of velocity and internal energy for the Riemann problem. The solid lines are traces of the M=200, k=1 solution from Figure 7b), and each open symbol is a nodal solution value obtained using $\vec{\nu}_\alpha^\gamma = \vec{\nu}_{opt}$. Interestingly, on the progression to quite coarse grids, the importance of $\vec{\nu}_\alpha^1 > 0$ on accuracy becomes diminished. The solid symbols in Figure 16 are the appropriate nodal values, obtained using $\vec{\nu}_\alpha^1 \equiv 0$, $\vec{\nu}_\alpha^2 = \nu\{3/4,2,1\}$ in the k=1 algorithm solution, at locations where the two solutions differed. As can be seen, the differences are truly negligible; hence, the two-dimensional Riemann solutions were all executed using $\vec{\nu}_\alpha^1 = 0$ and $\vec{\nu}_\alpha^2 = \vec{\nu}_{opt}^2$.

Several tests were conducted to evaluate application of vanishing gradient boundary conditions, on the transverse walls of the two-dimensional, inviscid flow Riemann shock tube. Figure 17 graphs the M=32x6, k=1 solution at t = 0.14154s, for the shock tube aligned parallel to the x_1-coordinate direction. As a consequence, the entire momentum solution is carried by

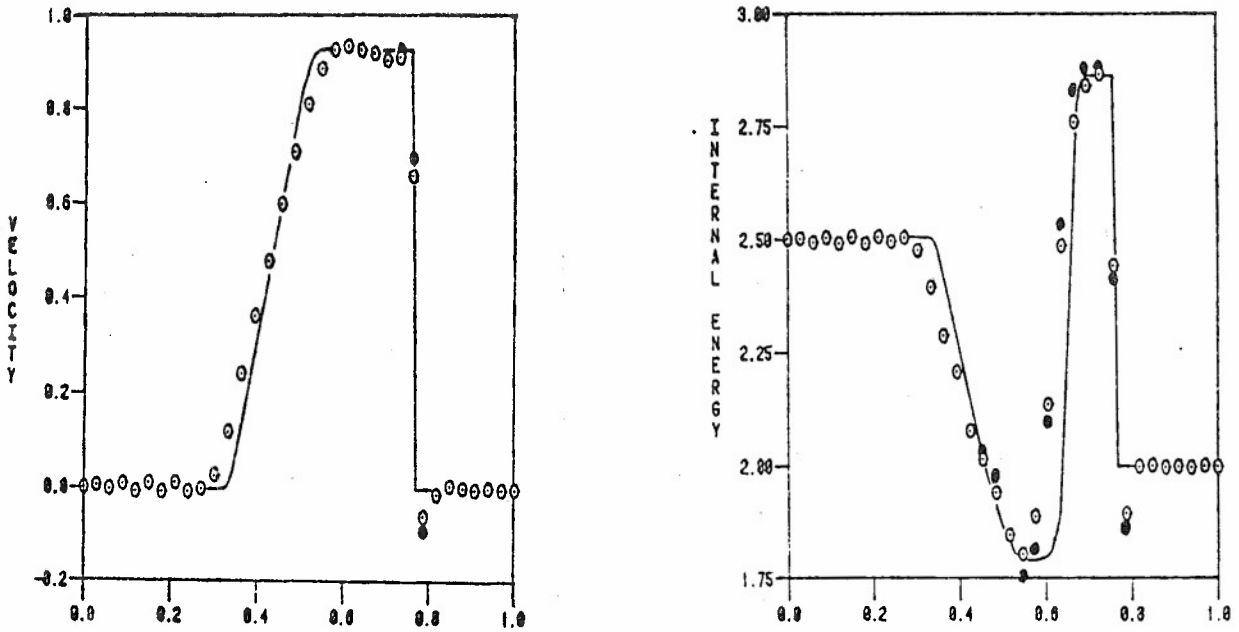
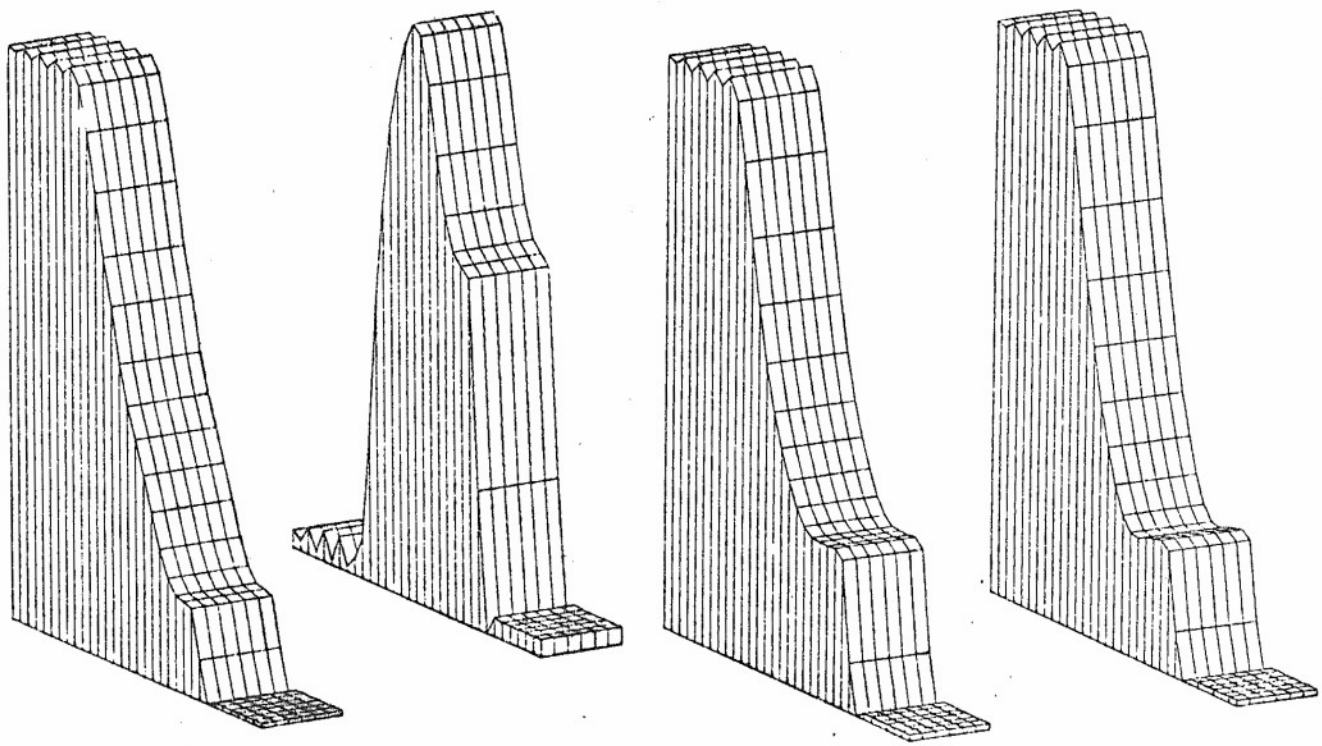


Figure 16. M=32x1, k=1 Finite Element Solution Parameters, Riemann Shock Tube, (——) M=200 Solution, (•) $\vec{\nu}_\alpha^1 = 0$, $\vec{\nu}_\alpha^2 = \vec{\nu}_{opt}$.



a) Density ρ b) Momentum m_1 c) Energy g d) Pressure p

Figure 17. M=32x6, k=1 Finite Element Solution, Two-Dimensional Riemann Shock Tube, t = 0.14154s, $\vec{\nu}_\alpha^1 = 0$, $\vec{\nu}_\alpha^2 = \nu\{3/4, 2, 1\}$.

m_1; indeed, m_2 was computed to equal zero to three significant digits, for the Newton convergence requirement set at $\varepsilon = 0.01$. The M=32x6 solution agrees almost exactly with the companion M=32x1, one-dimensional k=1 algorithm solution, see Figure 18, even to prediction of the small amplitude "$2\Delta x$" waves in the zero velocity region upstream of the rarefaction wave.

As shown in Figure 17, the tensor product algorithm can readily enforce a vanishing derivative transverse wall boundary condition without use of extraneous phantom cells exterior to R^2. A more demanding case corresponds to insertion of the shock tube diaphragm at an angle, which yields the requirement for the algorithm to predict a shock oblique to the mesh. The corresponding M=32x6, k=1 algorithm solution at t = 0.14154s is graphed in Figure 19. The algorithm again exhibits excellent solution fidelity, and experienced no difficulty in enforcing the gradient boundary constraint for steep solution gradients oblique to the walls. There occurs a slightly enhanced "$2\Delta x$" trashiness in the solution upper right corner, and a modest no-zero distribution for m_2 is also computed.

Figure 20 graphs the M=32x6, k=1 algorithm solution for the situation where the shock tube is aligned at an angle in the laboratory reference frame. As a consequence, since m_i is resolved into scalar components parallel to x_i, both momentum equations generate non-zero solutions. As can be seen, comparing Figures 17b) and 20b)-20c), the shock does not appear too noticeable in the m_i solutions. However, computation of the component of $\bar{u}_K$ parallel to η_1, which is the coordinate parallel to the shock tube axis, using equation 121 with {MJ} and {R}, confirms the existence of the shock, Figure 20e). Further, the prediction of the internal energy distribution, Figure 20f), is in good agreement with the "correct" solution, see Figure 16.

As the final test, the two-dimensional, aligned shock tube definition of Figure 17 was repeated as a viscous problem at $Re \equiv 10^5$, for an M=32x20 uniform discretization. The corresponding no-slip boundary conditions on the transverse walls are $m_1 = 0 = m_2$. The cold wall boundary condition $\varepsilon = \varepsilon_0$, equation 6, was defined for the energy equation, and a vanishing normal derivative applied for density. No boundary condition specifications are appropriate for p, σ_{ij} or q_j, since each is defined as an algebraic equation. The integration time step and Newton convergence requirement were maintained

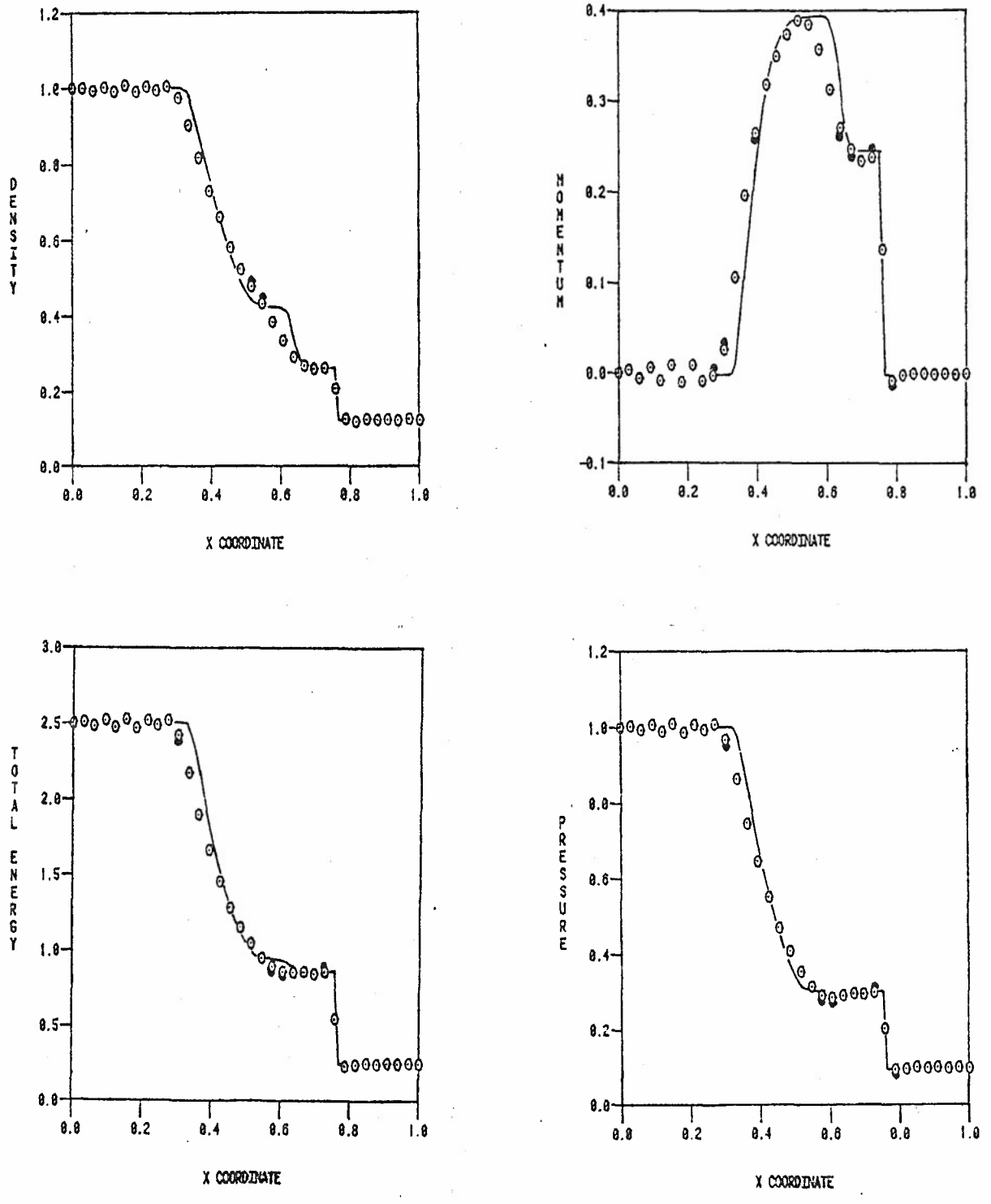


Figure 18. M=32x1, k=1 Finite Element Solution For Riemann Shock Tube, (——) M=200 Solution, (•) $\vec{\nu}_\alpha^1 = 0$, $\vec{\nu}_\alpha^2 = \vec{\nu}_{opt}$, t = 0.14154s.

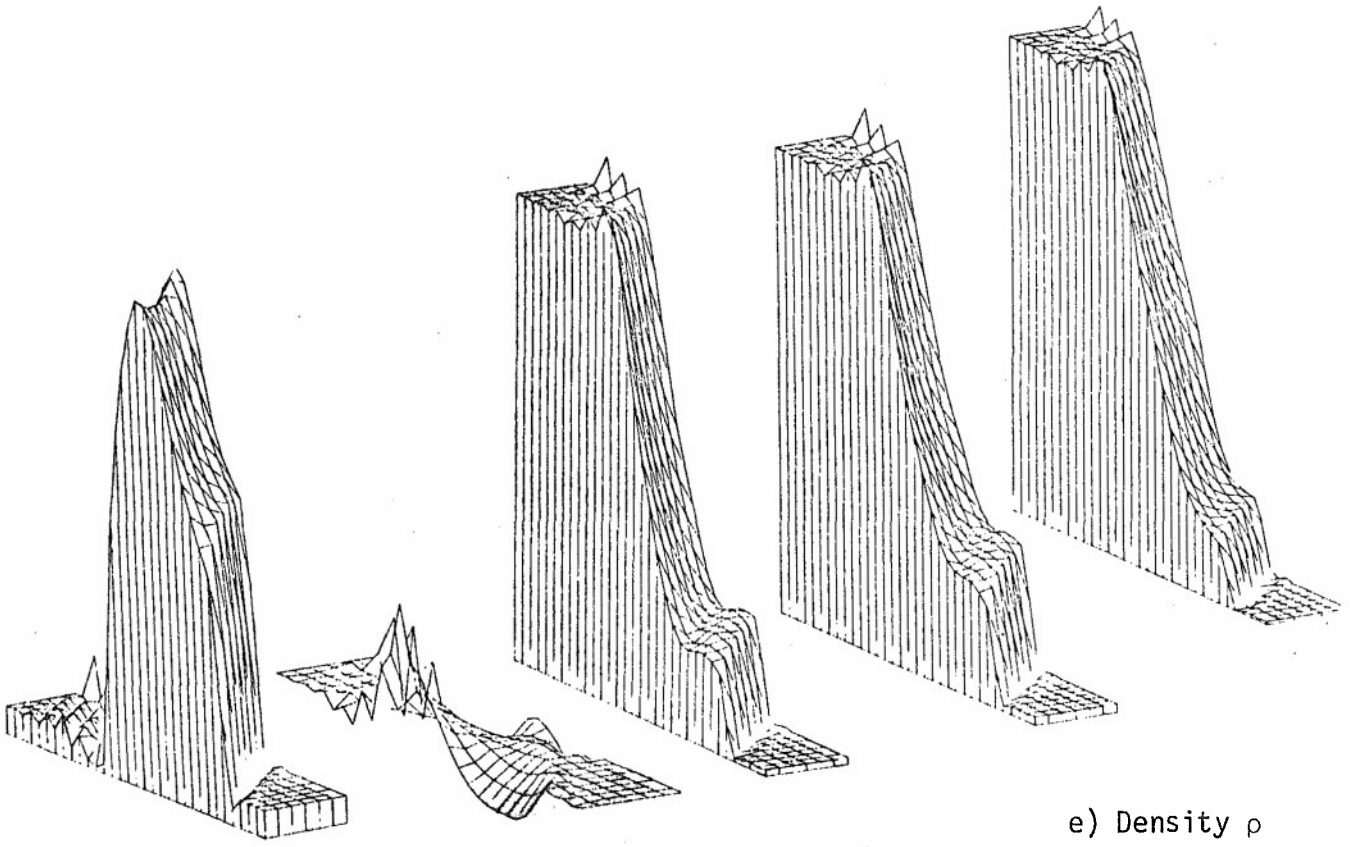
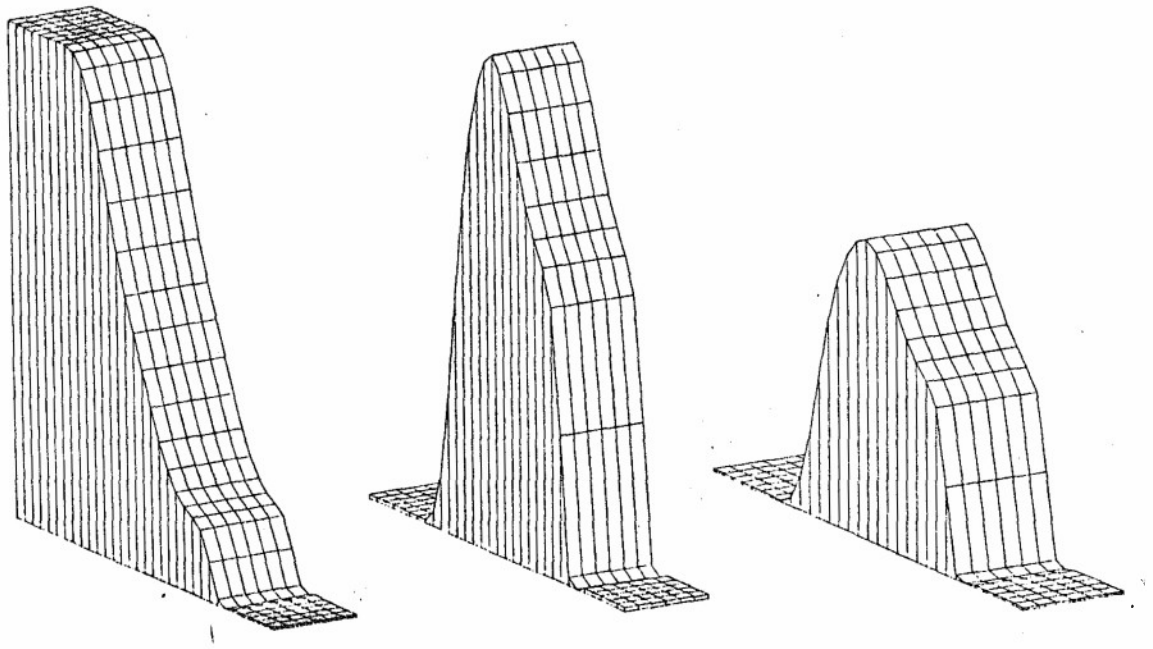
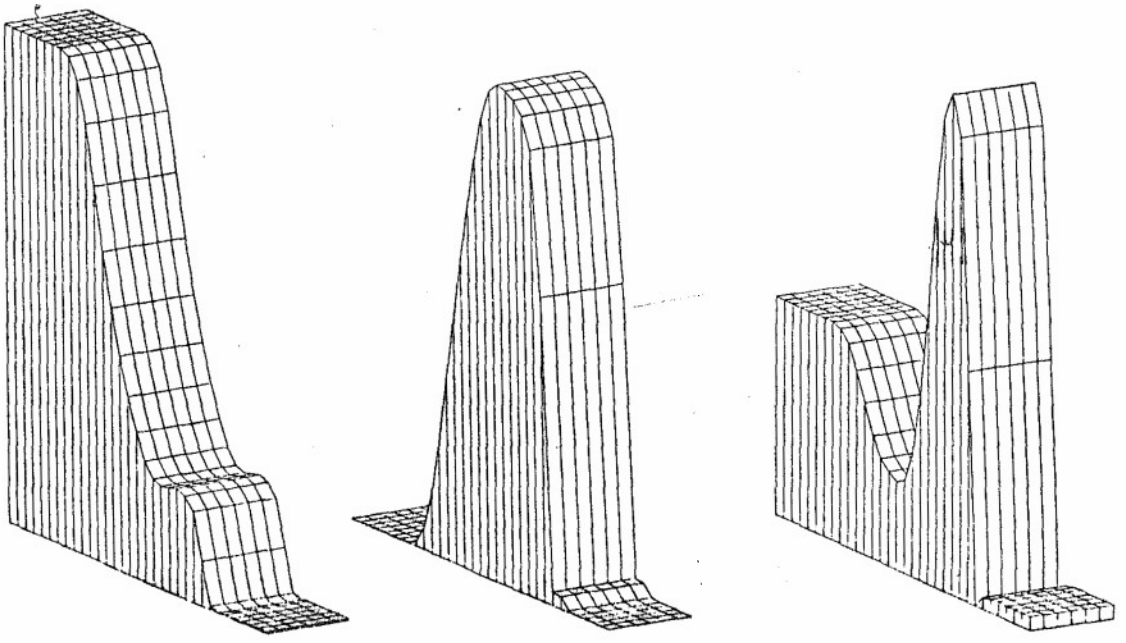


Figure 19. M=32x6, k=1 Finite Element Algorithm Solution, Two-Dimensional Riemann Shock Tube, Oblique Diaphragm, t = 0.14154s.

identical to the inviscid test case values. Figure 21 graphs the resulting k=1 algorithm solution at t = 0.14154s. In the centroidal region, away from the influence of the no-slip wall, the viscous solution agrees almost exactly with the M=32, k=1 one-dimensional inviscid solution, Figure 18. The influence of the viscous flow boundary conditions is clearly evident in all dependent variables. The growth of a viscous boundary layer behind the travelling shock is just visible in m_1. The solution for m_2 is highly oscillatory, but the peak value is only about 10% of the maximum m_1. The σ_{12} shear stress solution is sharply peaked at the wall, with a steep front adjacent to the shock. The solution is non-zero only in the vicinity of the wall, and exhibits the required skew-symmetries.

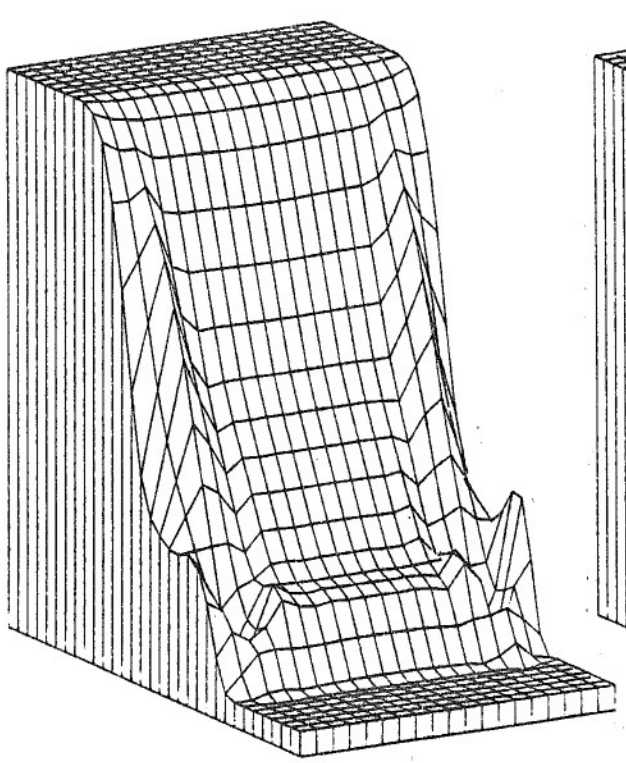


a) Density ρ b) Momentum m_1 c) Momentum m_2

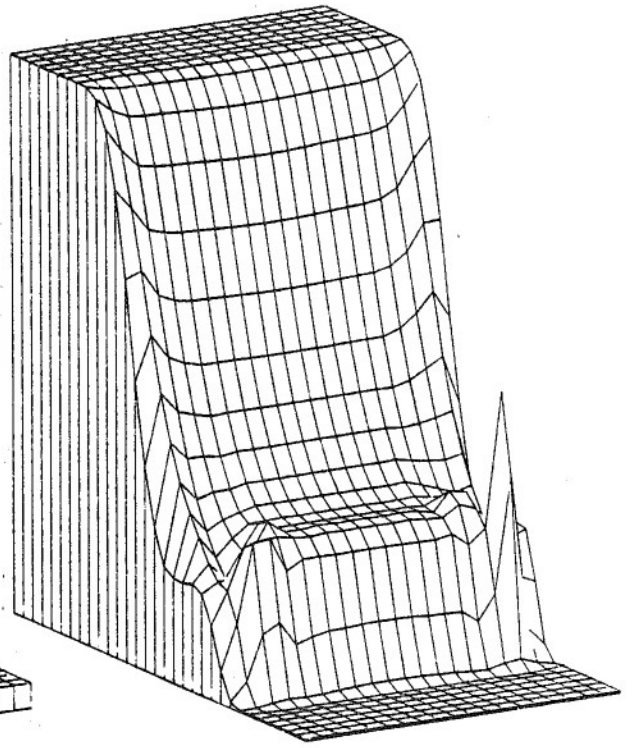


d) Pressure p e) Convection Velocity $\bar{u}_1$ f) Internal Energy

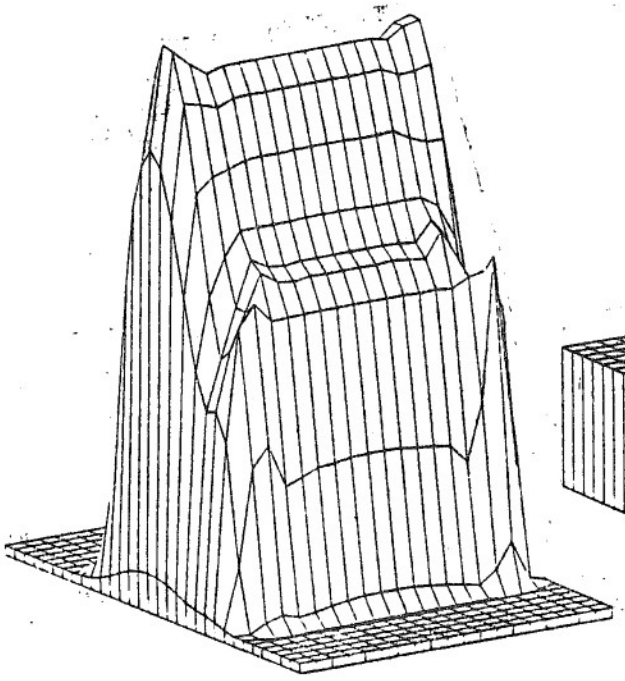
Figure 20. M=32x6, k=1 Finite Element Solution, Two-Dimensional Riemann Shock Tube, $\beta = 26°$, t = 0.14154s.



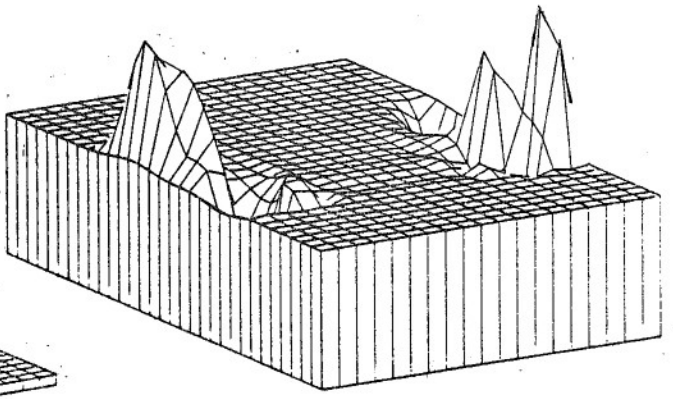
a) Density ρ



b) Energy

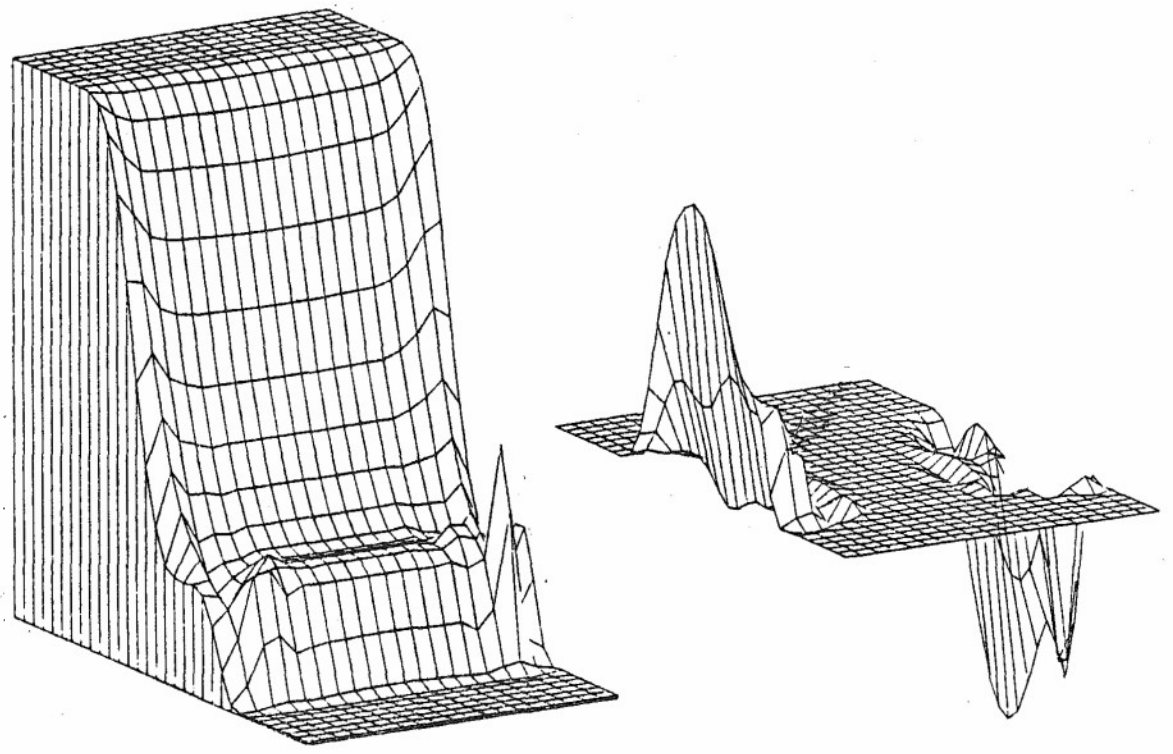


c) Momentum m_1



d) Momentum m_2

Figure 21. M=32x20, k=1 Finite Element Solution, Two-Dimensional Viscous Riemann Shock Tube, Re = 10^5, t = 0.14154s.



e) Pressure p

f) Shear Stress σ_{12}

Figure 21. M=32x20, k=1 Finite Element Solution, Two-Dimensional Viscous Riemann Shock Tube, Re = 10^5, t = 0.14154s, Concluded.

This test problem specification possessed ten dependent variables per nodal coordinate, plus five pieces of metric data, yielding approximately 10,000 nodal degrees of freedom to handle. The Newton convergence character was basically identical to the inviscid flow test case experience, following a few extra iterations at start-up. The solution employed 27 time steps to reach t_f, required 90 iterations, and utilized 70K words of central memory.

7. Three-Dimensional Problems, Metric Data

The metric data for three-dimensional problem definitions is generated in the manner discussed for R^2 in Section V.4. Denote the array of nodal coordinate triples of the discretization UR_e^3 as $\{XI\} \equiv \sum_e \{XI\}_e$, $1 \leq e \leq M$ and $1 \leq I \leq 3$. For the k=1 basis, recall Figure 1b), the finite element domain R_e^3 possesses only the vertex nodes $\{XI\}_e \equiv \{XI, YI, ZI, 1 \leq I \leq 8\}$. For the definition given in equation 116, the elements on the trace of $[\partial\eta_k/\partial x_j]$, evaluated at the centroid ($\eta_i = 0$, $1 \leq i \leq 3$) of R_e^3 are

$$\begin{aligned}
\{ETA11\}_e^T &= \tfrac{1}{4}\{Y41Z51,\ Y32Z62,\ Y32Z73,\ Y41Z84,\\
&\qquad Y85Z51,\ Y76Z62,\ Y76Z73,\ Y85Z84\}_e\\
\{ETA22\}_e^T &= \tfrac{1}{4}\{X21Z51,\ X21Z62,\ X34Z73,\ X34Z84,\\
&\qquad X65Z51,\ X65Z62,\ X78Z73,\ X78Z84\}_e\\
\{ETA33\}_e^T &= \tfrac{1}{4}\{X21Y41,\ X21Y32,\ X34Y32,\ X34Y41\\
&\qquad X65Y85,\ X65Y76,\ X78Y76,\ X78Y85\}_e
\end{aligned} \tag{125}$$

For a rectangular domain, $\{ETAKJ\}_e = \{0\}$ for $K \neq J$, and the notation in equation 125 is defined as, for example

$$Y41Z51 \equiv (Y4 - Y1)(Z5 - Z1) \tag{126}$$

In the same notation, the elements of $\{DET\}_e$, evaluated at the origin of the domain R_e^3, are

$$\begin{aligned}
\{DET\}_e^T = \frac{1}{64}\{&X21Y41Z51,\ X21Y32Z62,\ X34Y32Z73,\ X34Y41Z84,\\
&X65Y85Z51,\ X65Y76Z62,\ X78Y76Z73,\ X78Y85Z84\}_e
\end{aligned} \tag{127}$$

In the instance of R_e^3 being a rectangular parallelepiped, then $\{DET\}_e = \Delta_e\{1\}/64$, where Δ_e is the volume. For equations 125-127, nodes 1-4 lie in the lower plane, see Figure 1b, with node 1 in the lower left corner. Nodes 5-8 are also ordered counterclockwise, in the upper plane, with node 5 above node 1.

For the tensor product Jacobians, equations 66-71,

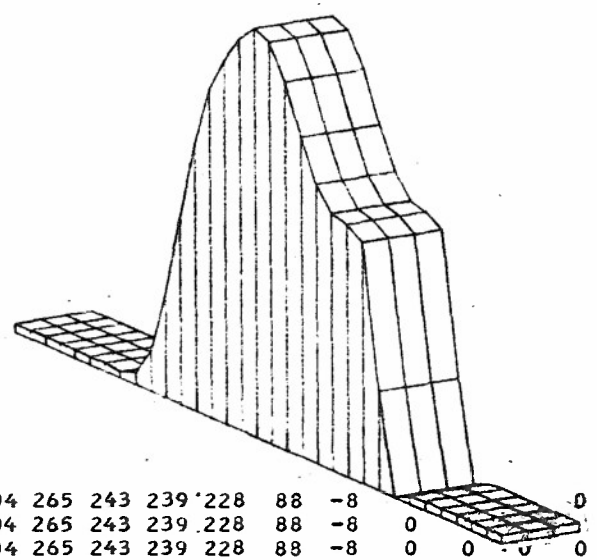
$$\{DETK\}_e = \tfrac{1}{2}\sqrt{(XKR - XKL)^2 + (YKR - YKR)^2 + (ZKR - ZKL)^2}\ \begin{Bmatrix}1\\1\end{Bmatrix} \qquad (128)$$

where R and L denote "right" and "left" respectively, for the specific alignment. As for R_e^2, the components of $\{ETAKJ\}_e$ for equations 66-71 are equivalent to the direction cosines of the local affine transformation between x_i and η_j, $1 \leq (i,j) \leq 3$. The definition for $\{UBARK\}_e$ remains unchanged from equation 124.

8. Three-Dimensional Riemann Shock Tube

The developed tensor product algorithm has been evaluated for the three-dimensional equivalent of the inviscid Riemann shock-tube problem, using the k=1 basis on an M = 32x4x4 uniform discretization of R^3. The principal requirement is to quantize performance of the tensor product Jacobian and the various vanishing normal gradient boundary conditions. The shock tube is aligned with the x_1 axis, and the dissipation parameter set is retained as defined for the two-dimensional simulation, $\vec{\nu}_\alpha^1 = 0$, $\vec{\nu}_\alpha^2 = \nu\{3/4,2,2,2,1\}$.

Each x_3 plane of the solution exhibits identical solution distributions, including at the nodal coordinates defined on the boundary ∂R of R^3. Further, each distribution is identical with the comparison two-dimensional solution, see Figure 17. For example, Figure 22 shows the actual print-out for $m_1(x_1,x_2)$, on the planes x_3 = constant, at t = 0.14154s. The total number of degrees of freedom for this case was approximately 3200, excluding the metric data. The solution employed 27 steps to reach t_f, required 56 iterations and utilized 64K words of central memory.



```
M1 VECTOR * 1000    1

  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0

M1 VECTOR * 1000    2

  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0

M1 VECTOR * 1000    3

  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0

M1 VECTOR * 1000    4

  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
  0   0   0   0   0   0   0  -2   4  31  93 174 253 312 347 373 390 384 351 304 265 243 239 228  88  -8   0   0   0   0
```

Figure 22. Program Print-Out For Three-Dimensional Riemann Shock Tube, Principal Momentum Component m_1, t = 0.14154s.

SECTION VI

CONCLUSIONS

An implicit, generalized-coordinates finite element numerical solution algorithm has been derived and evaluated for the three-dimensional Navier-Stokes equations. The theoretical basis utilizes a Galerkin-Weighted Residuals statement, rendering the semi-discrete error orthogonal to the finite element approximation subspace, augmented with a penalty constraint forcing orthogonality of the gradient of this error as well. As a consequence, the algorithm possesses highly phase selective dissipation mechanisms permitting accurate resolution of solutions exhibiting a high degree of non-smoothness. A Fourier stability analysis yields an estimate of the dissipation parameter set, which has been refined to enhance accuracy for a shocked flow prediction.

Multiple factors affecting solution accuracy, convergence and efficiency have been examined. The generalized coordinates framework directly facilitates matching of arbitrary surface descriptions of the solution domain for complete geometric versatility. This formulation as well permits establishment of the tensor matrix product approximation to the Newton iteration algorithm Jacobian, reducing by orders of magnitude the memory and CPU requirements for a multi-dimensional problem definition. The algebraic and constitutive equations, defining pressure, stress tensor and heat flux vector, are handled in an identical manner, yielding an overall consistency to the algorithm.

The results of a variety of numerical experiments have verified the basic attributes of the developed algorithm. However, with the limited time frame and funding for this project, the scope of numerical results that could be assessed has barely been touched. The theoretical construction of the algorithm does appear to contain, as a special case, the popular approximate-factorization finite difference algorithms in use today. However, the troublesome viscosity mixed derivatives are absent from the finite element formulation, yet the impact of this dependent variable choice has not been evaluated. Several other comparisons could and should be made, to more firmly quantify the desired algorithm comparisons.

REFERENCES

1. NASA Lewis Research Center Workshop on Advanced Computational Methods For Inlets, Diffusers, and Nozzles, Cleveland, OH, November, 1980.

2. MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," Technical Paper AIAA-69-354, 1969.

3. MacCormack, R. W., "An Efficient Explicit-Implicit-Characteristic Method for Solving the Compressible Navier-Stokes Equations," Proc. SIAM-AMS Symp. Comp. Fluid Dyn., New York, 1977.

4. MacCormack, R. W., "A Numerical Method For Solving the Equations of Compressible Viscous Flow," Technical Paper AIAA-81-0110, 1981.

5. Beam, R. M. and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," AIAA J., V. 16, 1978, pp. 393-402.

6. Briley, W. R. and McDonald, H., "Solution of the Multi-Dimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method," J. Comp. Phys., V. 24, 1977, pp. 372.

7. Thames, F. G., Thompson, J. F., Mastin, C. W., and Walker, R. L., "Numerical Solutions for Viscous and Potential Flow About Arbitrary Two-Dimensional Bodies Using Body-Fitted Coordinate Systems," J. Comp. Phys., V. 24, No. 1, 1977, pp. 245-273.

8. Proceedings, NASA Workshop On Numerical Grid Generation Techniques for Partial Differential Equations, NASA Report CP-2166, 1980.

9. Steger, J. L. and Pulliam, T. H., "An Implicit Finite Difference Code for Inviscid and Viscous Cascade Flow," Technical Paper AIAA 80-1427, 1980.

10. Steger, J. L., "Finite Difference Simulation of Compressible Flows," Presented at ASME-AIAA Symposium on Computers In Flow Predictions and Fluid Dynamics Experiments, ASME Winter Annual Meeting, Washington, DC, Nov. 1981.

11. Oden, J. T. and Reddy, J. N., An Introduction to the Mathematical Theory of Finite Elements, John Wiley, New York, 1976.

12. Oden, J. T., Finite Elements of Non-Linear Continua, McGraw-Hill, New York, 1972.

13. Shang, J. S., "Numerical Simulation of Wing-Fuselage Interference," Technical Paper AIAA-81-0048, 1981.

14. Baker, A. J., "Research on Numerical Algorithms for the Three-Dimensional Navier-Stokes Equations, I. Accuracy, Convergence and Efficiency," USAF Report AFFDL-TR-79-3141, 1979.

15. Baker, A. J., "Reseach On Numerical Algorithms for the Three-Dimensional Navier-Stokes Equations, II. Dissipative Finite Element," USAF Report AFWAL-TR-80-3157, 1981.

16. von Neumann, J. and Richtmyer, R. D., "A Method For the Numerical Calculation of Hydrodynamic Shocks," J. Appl. Phys., V. 21, 1950, pp. 232-237.

17. Gresho, P. M. and Lee, R. L., "Don't Suppress The Wiggles. They're Telling You Something!", Proc. ASME Sym. on Fin. El. Mtd. for Convection-Dominated Flows, Winter Annual Meeting, 1979.

18. Cebeci, T. and Smith, A. M. O., Analysis of Turbulent Boundary Layers, Academic Press, New York, 1974.

19. Hanjalic, K. and Launder, B. E., "A Reynolds Stress Model of Turbulence and its Application to Thin Shear Flows," J. Fluid Mech. V. 52, Pt. 4, 1972, pp. 609-638.

20. Baker, A. J. and Orzechowski, J. A., "A Continuity-Constraint Finite Element Algorithm for Three-Dimensional Parabolic Flow Prediction," Joint ASME-AIAA Symposium on Computers in Flow Prediction and Experiments, Nov. 1981.

21. Prenter, P. M., Splines and Variational Methods, John Wiley, New York, 1975.

22. Halmos, P. R., Finite Dimensional Vector Spaces, Van Nostrand, Princeton, NJ, 1958.

23. Baker, A. J., Finite Element Computational Fluid Mechanics, McGraw-Hill-Hemisphere, New York, 1983 (tentative).

24. Strang, G. and Fix, G. J., An Analysis of the Finite Element Method, Prentice Hall, Englewood Cliffs, NJ, 1973.

25. Baker, A. J. and Soliman, M. O., "Utility of a Finite Element Solution Algorithm for Initial-Value Problems," J. Comp. Phys., V. 32, No. 3, 1979, pp. 289-324.

26. Soliman, M. O. and Baker, A. J., "Accuracy and Convergence of a Finite Element Algorithm for Laminar Boundary Layer Flow," J. Computers and Fluids, V. 9, 1981, pp. 43-62.

27. Soliman, M. O. and Baker, A. J., "Accuracy and Convergence of a Finite Element Algorithm for Turbulent Boundary Layer Flow," Comp. Mtd. Appl. Mech. & Engr., V. 28, 1981, pp. 81-102.

28. Raymond, W. H. and Gardner, A., "Selective Damping in a Galerkin Method for Solving Wave Problems with Variable Grids," Monthly Weather Rev., Vol. 104, 1976, pp. 1583-1590.

29. Pepper, D. W. and Cooper, R. E., "Numerical Solution of REcirculating Flow by a Simple Finite Element Recursion Relation," J. Comp. and Fluids, V. 8, 1980, pp. 213-223.

30. Shapiro, A. H., The Dynamics And Thermodynamics of Compressible Fluid Flow, V. II, Ronald Press, NY, 1953.

31. Van Leer, B., "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," J. Comp. Phys., V. 32, 1979, pp. 101-136.

32. Zalesak, S. T., "High Order ZIP Differencing of Convective Terms," NRL Report No. 4218, May, 1980, J. Comp. Phys., to appear.

33. Sod, G. A., "A Survey of Several Finite Difference Methods for Systems of Non-Linear Hyperbolic Conservation Laws," J. Comp. Phys., V. 27, pp. 1-31, 1978.

APPENDIX A

FINITE ELEMENT ALGORITHM HYPERMATRICES
LINEAR AND QUADRATIC BASIS ON ONE DIMENSIONAL SPACE

1. Linear Finite Element Formulation

$$\{A10\} = \frac{1}{2}\begin{Bmatrix}1\\1\end{Bmatrix}$$

$$[A200] = \frac{1}{6}\begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}$$

$$[A201] = \frac{1}{2\Delta_e}\begin{bmatrix}-1 & 1\\-1 & 1\end{bmatrix}$$

$$[A210] = \frac{1}{2\Delta_e}\begin{bmatrix}-1 & -1\\1 & 1\end{bmatrix}$$

$$[A3000] = \frac{1}{12}\begin{bmatrix}\begin{Bmatrix}3\\1\end{Bmatrix}\begin{Bmatrix}1\\1\end{Bmatrix}\\ \begin{Bmatrix}1\\1\end{Bmatrix}\begin{Bmatrix}1\\3\end{Bmatrix}\end{bmatrix}$$

$$[A3001] = \frac{1}{6\Delta_e}\begin{bmatrix}\begin{Bmatrix}-2\\-1\end{Bmatrix}\begin{Bmatrix}2\\1\end{Bmatrix}\\ \begin{Bmatrix}-1\\-2\end{Bmatrix}\begin{Bmatrix}1\\2\end{Bmatrix}\end{bmatrix}$$

$$[A3010] = \frac{1}{6\Delta_e}\begin{bmatrix} \begin{Bmatrix} -2 \\ -1 \end{Bmatrix} & \begin{Bmatrix} -1 \\ -2 \end{Bmatrix} \\ \begin{Bmatrix} 2 \\ 1 \end{Bmatrix} & \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} \end{bmatrix}$$

$$[A3100] = \frac{1}{6\Delta_e}\begin{bmatrix} \begin{Bmatrix} -2 \\ 2 \end{Bmatrix} & \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \\ \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} & \begin{Bmatrix} -2 \\ 2 \end{Bmatrix} \end{bmatrix}$$

$$[A3011] = \frac{1}{2\Delta_e^2}\begin{bmatrix} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} & \begin{Bmatrix} -1 \\ -1 \end{Bmatrix} \\ \begin{Bmatrix} -1 \\ -1 \end{Bmatrix} & \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \end{bmatrix}$$

$$[A3110] = \frac{1}{2\Delta_e^2}\begin{bmatrix} \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} & \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} \\ \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} & \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \end{bmatrix}$$

$$[A40000] = \frac{1}{60}\begin{bmatrix} \begin{bmatrix} 12 & 3 \\ 3 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \\ \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} & \begin{bmatrix} 2 & 3 \\ 3 & 12 \end{bmatrix} \end{bmatrix}$$

$$[A40001] = \frac{1}{12\Delta_e}\begin{bmatrix} \begin{bmatrix} -3 & 3 \\ -1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \\ \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & 1 \\ -3 & 3 \end{bmatrix} \end{bmatrix}$$

$$[A40010] = \frac{1}{12\Delta_e}\begin{bmatrix} \begin{bmatrix} -3 & -1 \\ -1 & -1 \end{bmatrix} & \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix} \\ \begin{bmatrix} -1 & -1 \\ -1 & -3 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix} \end{bmatrix}$$

$$[A41000] = \frac{1}{12\Delta_e}\begin{bmatrix} \begin{bmatrix} -3 & -1 \\ 3 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \\ \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -3 \\ 1 & 3 \end{bmatrix} \end{bmatrix}$$

$$[A40011] = \frac{1}{6\Delta_e}\begin{bmatrix} \begin{bmatrix} 2 & -2 \\ 1 & -1 \end{bmatrix} & \begin{bmatrix} -2 & 2 \\ -1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & -1 \\ 2 & -2 \end{bmatrix} & \begin{bmatrix} -1 & 1 \\ -2 & 2 \end{bmatrix} \end{bmatrix}$$

$$[A40110] = \frac{1}{6\Delta_e^2}\begin{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} & \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix} \\ \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix} & \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \end{bmatrix}$$

$$[A41100] = \frac{1}{6\Delta_e^2}\begin{bmatrix} \begin{bmatrix} 2 & 1 \\ -2 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 2 \\ -1 & -2 \end{bmatrix} \\ \begin{bmatrix} -2 & -1 \\ 2 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -2 \\ 1 & 2 \end{bmatrix} \end{bmatrix}$$

$$[A40101] = \frac{1}{6\Delta_e^2}\begin{bmatrix} \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} & \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} & \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} \end{bmatrix}$$

2. Quadratic Finite Element Formulation

$$\{A10\} = \frac{1}{6}\begin{Bmatrix} 1 \\ 4 \\ 1 \end{Bmatrix}$$

$$[A200] = \frac{1}{30}\begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix}$$

$$[A201] = \frac{1}{6\Delta_e}\begin{bmatrix} -3 & 4 & -1 \\ -4 & 0 & 4 \\ 1 & -4 & 3 \end{bmatrix}$$

$$[A210] = \frac{1}{6\Delta_e}\begin{bmatrix} -3 & -4 & 1 \\ 4 & 0 & 4 \\ -1 & 4 & 3 \end{bmatrix}$$

$$[A3000] = \frac{1}{420}\begin{bmatrix} \begin{Bmatrix} 39 \\ 20 \\ -3 \end{Bmatrix} & \begin{Bmatrix} 20 \\ 16 \\ -8 \end{Bmatrix} & \begin{Bmatrix} -3 \\ -8 \\ -8 \end{Bmatrix} \\ \begin{Bmatrix} 20 \\ 16 \\ -8 \end{Bmatrix} & \begin{Bmatrix} 16 \\ 192 \\ 16 \end{Bmatrix} & \begin{Bmatrix} -8 \\ 16 \\ 20 \end{Bmatrix} \\ \begin{Bmatrix} -3 \\ -8 \\ -3 \end{Bmatrix} & \begin{Bmatrix} -8 \\ 16 \\ 20 \end{Bmatrix} & \begin{Bmatrix} -3 \\ 20 \\ 39 \end{Bmatrix} \end{bmatrix}$$

$$[A3001] = \frac{1}{90A_e}\begin{bmatrix} \begin{Bmatrix} -30 \\ -18 \\ 3 \end{Bmatrix} & \begin{Bmatrix} 36 \\ 24 \\ 0 \end{Bmatrix} & \begin{Bmatrix} -6 \\ -6 \\ -3 \end{Bmatrix} \\ \begin{Bmatrix} -18 \\ -48 \\ 6 \end{Bmatrix} & \begin{Bmatrix} 24 \\ 0 \\ -24 \end{Bmatrix} & \begin{Bmatrix} -6 \\ 48 \\ 18 \end{Bmatrix} \\ \begin{Bmatrix} 3 \\ 6 \\ 6 \end{Bmatrix} & \begin{Bmatrix} 0 \\ -24 \\ -36 \end{Bmatrix} & \begin{Bmatrix} -3 \\ 18 \\ 30 \end{Bmatrix} \end{bmatrix}$$

$$[A3010] = \frac{1}{30\Delta_e}\begin{bmatrix} \begin{Bmatrix}-10\\-6\\1\end{Bmatrix} & \begin{Bmatrix}-6\\-16\\2\end{Bmatrix} & \begin{Bmatrix}1\\2\\2\end{Bmatrix} \\ \begin{Bmatrix}12\\8\\0\end{Bmatrix} & \begin{Bmatrix}8\\0\\-8\end{Bmatrix} & \begin{Bmatrix}0\\-8\\-12\end{Bmatrix} \\ \begin{Bmatrix}-2\\-2\\-1\end{Bmatrix} & \begin{Bmatrix}-2\\16\\6\end{Bmatrix} & \begin{Bmatrix}-1\\6\\10\end{Bmatrix} \end{bmatrix}$$

$$[A3100] = \frac{1}{30\Delta_e}\begin{bmatrix} \begin{Bmatrix}-10\\12\\-2\end{Bmatrix} & \begin{Bmatrix}-6\\8\\-2\end{Bmatrix} & \begin{Bmatrix}1\\0\\-1\end{Bmatrix} \\ \begin{Bmatrix}-6\\8\\-2\end{Bmatrix} & \begin{Bmatrix}-16\\0\\16\end{Bmatrix} & \begin{Bmatrix}2\\-8\\6\end{Bmatrix} \\ \begin{Bmatrix}1\\0\\-1\end{Bmatrix} & \begin{Bmatrix}2\\-8\\6\end{Bmatrix} & \begin{Bmatrix}2\\-12\\10\end{Bmatrix} \end{bmatrix}$$

$$[A3011] = \frac{1}{30\Delta_e^2}\begin{bmatrix} \begin{Bmatrix}37\\36\\-3\end{Bmatrix} & \begin{Bmatrix}-44\\-32\\-4\end{Bmatrix} & \begin{Bmatrix}7\\-4\\7\end{Bmatrix} \\ \begin{Bmatrix}-44\\-32\\-4\end{Bmatrix} & \begin{Bmatrix}48\\64\\48\end{Bmatrix} & \begin{Bmatrix}-4\\-32\\-44\end{Bmatrix} \\ \begin{Bmatrix}7\\-4\\7\end{Bmatrix} & \begin{Bmatrix}-4\\-32\\-44\end{Bmatrix} & \begin{Bmatrix}-3\\36\\37\end{Bmatrix} \end{bmatrix}$$

$$[A3110] = \frac{1}{30\Delta_e^2}\begin{bmatrix} \begin{Bmatrix} 37 \\ -44 \\ 7 \end{Bmatrix} & \begin{Bmatrix} 36 \\ -32 \\ -4 \end{Bmatrix} & \begin{Bmatrix} -3 \\ -4 \\ 7 \end{Bmatrix} \\ \begin{Bmatrix} -44 \\ 48 \\ -4 \end{Bmatrix} & \begin{Bmatrix} -32 \\ 64 \\ -32 \end{Bmatrix} & \begin{Bmatrix} -4 \\ 48 \\ 44 \end{Bmatrix} \\ \begin{Bmatrix} 7 \\ 4 \\ -3 \end{Bmatrix} & \begin{Bmatrix} -4 \\ -32 \\ 36 \end{Bmatrix} & \begin{Bmatrix} 7 \\ -44 \\ 37 \end{Bmatrix} \end{bmatrix}$$

$$[A40000] = \frac{1}{630}\begin{bmatrix} \begin{bmatrix} 46 & 16 & -3 \\ 16 & 16 & -2 \\ -3 & -2 & -1 \end{bmatrix} & \begin{bmatrix} 16 & 16 & -2 \\ 16 & 16 & -8 \\ -2 & -8 & -2 \end{bmatrix} & \begin{bmatrix} -3 & -2 & 1 \\ -2 & -8 & -2 \\ 1 & -2 & -3 \end{bmatrix} \\ \begin{bmatrix} 16 & 16 & -2 \\ 16 & 16 & -8 \\ -2 & -8 & -2 \end{bmatrix} & \begin{bmatrix} 16 & 16 & -8 \\ 16 & 256 & 16 \\ -8 & 16 & 16 \end{bmatrix} & \begin{bmatrix} -2 & -8 & -2 \\ -8 & 16 & 16 \\ -2 & 16 & 16 \end{bmatrix} \\ \begin{bmatrix} -3 & -2 & 1 \\ -2 & -8 & -2 \\ 1 & -2 & -3 \end{bmatrix} & \begin{bmatrix} -2 & -8 & -2 \\ -8 & 16 & 16 \\ -2 & 16 & 16 \end{bmatrix} & \begin{bmatrix} 1 & -2 & -3 \\ -2 & 16 & 16 \\ -3 & 16 & 46 \end{bmatrix} \end{bmatrix}$$

$$[A40001] = \frac{1}{420\Delta_e}\begin{bmatrix} \begin{bmatrix} -105 & 132 & -27 \\ -44 & 48 & -4 \\ 9 & -12 & 3 \end{bmatrix} & \begin{bmatrix} -44 & 48 & -4 \\ -48 & 64 & -16 \\ 8 & 0 & -8 \end{bmatrix} & \begin{bmatrix} 9 & -12 & 3 \\ 8 & 0 & -8 \\ -3 & 12 & -9 \end{bmatrix} \\ \begin{bmatrix} -44 & 48 & -4 \\ -48 & 64 & -16 \\ 8 & 0 & -8 \end{bmatrix} & \begin{bmatrix} -48 & 64 & -16 \\ -192 & 0 & 192 \\ 16 & -64 & 48 \end{bmatrix} & \begin{bmatrix} 8 & 0 & -8 \\ 16 & -64 & 48 \\ 4 & -48 & 44 \end{bmatrix} \\ \begin{bmatrix} 9 & -12 & 3 \\ 8 & 0 & -8 \\ -3 & 12 & -9 \end{bmatrix} & \begin{bmatrix} 8 & 0 & -8 \\ 16 & -64 & 48 \\ 4 & -48 & 44 \end{bmatrix} & \begin{bmatrix} -3 & 12 & -9 \\ 4 & -48 & 44 \\ 27 & -132 & 105 \end{bmatrix} \end{bmatrix}$$

$$[A40010] = \frac{1}{420\Delta_e}\begin{bmatrix}
\begin{bmatrix} -105 & -44 & 9 \\ -44 & -48 & 8 \\ 9 & 8 & -3 \end{bmatrix} & \begin{bmatrix} 132 & 48 & -12 \\ 48 & 64 & 0 \\ -12 & 0 & 12 \end{bmatrix} & \begin{bmatrix} -27 & -4 & 3 \\ -4 & -16 & -8 \\ 3 & -8 & -9 \end{bmatrix} \\
\begin{bmatrix} -44 & -48 & 8 \\ -48 & -192 & 16 \\ 8 & 16 & 4 \end{bmatrix} & \begin{bmatrix} 48 & 64 & 0 \\ 64 & 0 & -64 \\ 0 & -64 & -48 \end{bmatrix} & \begin{bmatrix} -4 & -16 & -8 \\ -16 & 192 & 48 \\ -8 & 48 & 44 \end{bmatrix} \\
\begin{bmatrix} 9 & 8 & -3 \\ 8 & 16 & 4 \\ -3 & 4 & 27 \end{bmatrix} & \begin{bmatrix} -12 & 0 & 12 \\ 0 & -64 & -48 \\ 12 & -48 & -132 \end{bmatrix} & \begin{bmatrix} 3 & -8 & -9 \\ -8 & 48 & 44 \\ -8 & 44 & 105 \end{bmatrix}
\end{bmatrix}$$

$$[A41000] = \frac{1}{420\Delta_e}\begin{bmatrix}
\begin{bmatrix} -105 & -44 & 9 \\ -44 & -48 & 8 \\ 9 & 8 & -3 \end{bmatrix} & \begin{bmatrix} -44 & -48 & 8 \\ -48 & -192 & 16 \\ 8 & 16 & 4 \end{bmatrix} & \begin{bmatrix} 9 & 8 & -3 \\ 8 & 16 & 4 \\ -3 & 4 & 27 \end{bmatrix} \\
\begin{bmatrix} 132 & 48 & -12 \\ 48 & 64 & 0 \\ -12 & 0 & 12 \end{bmatrix} & \begin{bmatrix} 48 & 64 & 0 \\ 64 & 0 & -64 \\ 0 & 64 & -48 \end{bmatrix} & \begin{bmatrix} -12 & 0 & 12 \\ 0 & -64 & -48 \\ 12 & -48 & -132 \end{bmatrix} \\
\begin{bmatrix} -27 & -4 & 3 \\ -4 & -16 & -8 \\ 3 & -8 & -9 \end{bmatrix} & \begin{bmatrix} -4 & -16 & -8 \\ -16 & 192 & 48 \\ -8 & 48 & 44 \end{bmatrix} & \begin{bmatrix} 3 & -8 & -9 \\ -8 & 48 & 44 \\ -9 & 44 & 105 \end{bmatrix}
\end{bmatrix}$$

$$[A40011] = \frac{1}{210\Delta_e^2}\begin{bmatrix}
\begin{bmatrix} 184 & -228 & 44 \\ 94 & -104 & 10 \\ -19 & 24 & -5 \end{bmatrix} & \begin{bmatrix} -228 & 288 & -60 \\ -104 & 96 & 8 \\ 24 & -48 & 24 \end{bmatrix} & \begin{bmatrix} 44 & -60 & 16 \\ 10 & 8 & -18 \\ -5 & 24 & -19 \end{bmatrix} \\
\begin{bmatrix} 94 & -104 & 10 \\ 176 & -128 & -48 \\ -18 & 8 & 10 \end{bmatrix} & \begin{bmatrix} -104 & 96 & 8 \\ -128 & 256 & -128 \\ 8 & 96 & -104 \end{bmatrix} & \begin{bmatrix} 10 & 8 & -18 \\ -48 & -128 & 176 \\ 10 & -104 & 94 \end{bmatrix} \\
\begin{bmatrix} -19 & 24 & -5 \\ -18 & 8 & 10 \\ 16 & -60 & 44 \end{bmatrix} & \begin{bmatrix} 24 & -48 & 24 \\ 8 & 96 & -104 \\ -60 & 288 & -228 \end{bmatrix} & \begin{bmatrix} -5 & 24 & -19 \\ 10 & -104 & 94 \\ 44 & -228 & 184 \end{bmatrix}
\end{bmatrix}$$

$$[A40110] = \frac{1}{210\Delta_e^2}\begin{bmatrix}
\begin{bmatrix} 184 & 94 & -19 \\ -228 & -104 & 24 \\ 44 & 10 & -5 \end{bmatrix} & \begin{bmatrix} -228 & -104 & 24 \\ 288 & 96 & -48 \\ -60 & 8 & 24 \end{bmatrix} & \begin{bmatrix} 44 & 10 & -5 \\ -60 & 8 & 24 \\ 16 & -18 & -19 \end{bmatrix} \\
\begin{bmatrix} 94 & 176 & -18 \\ -104 & -128 & 8 \\ 10 & -48 & 10 \end{bmatrix} & \begin{bmatrix} -104 & -128 & 8 \\ 96 & 256 & 96 \\ 8 & -128 & -104 \end{bmatrix} & \begin{bmatrix} 10 & -48 & 10 \\ 8 & -128 & -104 \\ -18 & 176 & 94 \end{bmatrix} \\
\begin{bmatrix} -19 & -18 & 16 \\ 24 & 8 & -60 \\ -5 & 10 & 44 \end{bmatrix} & \begin{bmatrix} 24 & 8 & -60 \\ -48 & 96 & 288 \\ 24 & -104 & -228 \end{bmatrix} & \begin{bmatrix} -5 & 10 & 44 \\ 24 & -104 & -228 \\ -19 & 94 & 184 \end{bmatrix}
\end{bmatrix}$$

$$[A41100] = \frac{1}{210\Delta_e^2}\begin{bmatrix}
\begin{bmatrix} 184 & 94 & -19 \\ -228 & -104 & 24 \\ 44 & 10 & -5 \end{bmatrix} & \begin{bmatrix} 94 & 176 & -18 \\ -104 & -128 & 8 \\ 10 & -48 & 10 \end{bmatrix} & \begin{bmatrix} -19 & -18 & 16 \\ 24 & 8 & -60 \\ -5 & 10 & 44 \end{bmatrix} \\
\begin{bmatrix} -228 & -104 & 24 \\ 288 & 96 & -48 \\ -60 & 8 & 24 \end{bmatrix} & \begin{bmatrix} -104 & -128 & 8 \\ 96 & 256 & 96 \\ 8 & -128 & -104 \end{bmatrix} & \begin{bmatrix} 24 & 8 & -60 \\ -48 & 96 & 288 \\ 24 & -104 & -228 \end{bmatrix} \\
\begin{bmatrix} 44 & 10 & -5 \\ -60 & 8 & 24 \\ 16 & -18 & -19 \end{bmatrix} & \begin{bmatrix} 10 & -48 & 10 \\ 8 & -128 & -104 \\ -18 & 176 & 94 \end{bmatrix} & \begin{bmatrix} -5 & 10 & 44 \\ 24 & -104 & -228 \\ -19 & 94 & 184 \end{bmatrix}
\end{bmatrix}$$

$$[A40101] = \frac{1}{210\Delta_e^2}\begin{bmatrix}
\begin{bmatrix} 184 & -228 & 44 \\ -228 & 288 & -60 \\ 44 & -60 & 16 \end{bmatrix} & \begin{bmatrix} 94 & -104 & 10 \\ -104 & 96 & 8 \\ 10 & 8 & -18 \end{bmatrix} & \begin{bmatrix} -19 & 24 & -5 \\ 24 & -48 & 24 \\ -5 & 24 & -19 \end{bmatrix} \\
\begin{bmatrix} 94 & -104 & 10 \\ -104 & 96 & 8 \\ 10 & 8 & -18 \end{bmatrix} & \begin{bmatrix} 176 & -128 & -48 \\ -128 & 256 & -128 \\ -48 & -128 & 176 \end{bmatrix} & \begin{bmatrix} -18 & 8 & 10 \\ 8 & 96 & -104 \\ 10 & -104 & 94 \end{bmatrix} \\
\begin{bmatrix} -19 & 24 & -5 \\ 24 & -48 & 24 \\ -5 & 24 & -19 \end{bmatrix} & \begin{bmatrix} -18 & 8 & 10 \\ 8 & 96 & -104 \\ 10 & -104 & 94 \end{bmatrix} & \begin{bmatrix} 16 & -60 & 44 \\ -60 & 288 & -228 \\ 44 & -228 & 184 \end{bmatrix}
\end{bmatrix}$$

APPENDIX B

FINITE ELEMENT ALGORITHM HYPERMATRICES

BI-LINEAR TENSOR PRODUCT BASIS ON TWO-DIMENSIONAL SPACE

$$\{B10\} = \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix}$$

$$[B200] = \frac{1}{9}\begin{bmatrix} 4 & 2 & 1 & 2 \\ & 4 & 2 & 1 \\ & & 4 & 2 \\ (sym) & & & 4 \end{bmatrix}$$

$$[B3000] = \frac{1}{36}\begin{bmatrix} 9 & 3 & 1 & 3 \\ 3 & 3 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 3 \\ 3 & 3 & 1 & 1 \\ 3 & 9 & 3 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 3 & 3 & 1 \\ 1 & 3 & 9 & 3 \\ 1 & 1 & 3 & 3 \\ 3 & 1 & 1 & 3 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 3 & 3 \\ 3 & 1 & 3 & 9 \end{bmatrix}$$

$$[B3001] = \frac{1}{36}\begin{bmatrix} -6 & 6 & 2 & -2 \\ -3 & 3 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -2 & 2 & 2 & -2 \\ -3 & 3 & 1 & -1 \\ -6 & 6 & 2 & -2 \\ -2 & 2 & 2 & -2 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -2 & 2 & 2 & -2 \\ -2 & 2 & 6 & -6 \\ -1 & 1 & 3 & -3 \\ -2 & 2 & 2 & -2 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 3 & -3 \\ -2 & 2 & 6 & -6 \end{bmatrix}$$

$$[B3010] = \frac{1}{36}\begin{bmatrix} -6 & -3 & -1 & -2 \\ -3 & -6 & -2 & -1 \\ -1 & -2 & -2 & -1 \\ -2 & -1 & -1 & -2 \\ 6 & 3 & 1 & 2 \\ 3 & 6 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 1 & 2 \\ 2 & 1 & 1 & 2 \\ 1 & 2 & 2 & 1 \\ 1 & 2 & 6 & 3 \\ 2 & 1 & 3 & 6 \\ -2 & -1 & -1 & -2 \\ -1 & -2 & -2 & -1 \\ -1 & -2 & -6 & -3 \\ -2 & -1 & -3 & -6 \end{bmatrix}$$

$$[B3002] = \frac{1}{36}\begin{bmatrix} -6 & -2 & 2 & 6 \\ -2 & -2 & 2 & 2 \\ -1 & -1 & 1 & 1 \\ -3 & -1 & 1 & 3 \\ -2 & -2 & 2 & 2 \\ -2 & -6 & 6 & 2 \\ -1 & -3 & 3 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -3 & 3 & 1 \\ -2 & -6 & 6 & 2 \\ -2 & -2 & 2 & 2 \\ -3 & -1 & 1 & 3 \\ -1 & -1 & 1 & 1 \\ -2 & -2 & 2 & 2 \\ -6 & -2 & 2 & 6 \end{bmatrix}$$

$$[B3020] = \frac{1}{36}\begin{bmatrix} -6 & -2 & -1 & -3 \\ -2 & -2 & -1 & -1 \\ -1 & -1 & -2 & -2 \\ -3 & -1 & -2 & -6 \\ -2 & -2 & -1 & -1 \\ -2 & -6 & -3 & -1 \\ -1 & -3 & -6 & -2 \\ -1 & -1 & -2 & -2 \\ 2 & 2 & 1 & 1 \\ 2 & 6 & 3 & 1 \\ 1 & 3 & 6 & 2 \\ 1 & 1 & 2 & 2 \\ 6 & 2 & 1 & 3 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 2 \\ 3 & 1 & 2 & 6 \end{bmatrix}$$

$$[B3011] = \frac{1}{36}\begin{bmatrix} 4 & -4 & -1 & 1 \\ 4 & -4 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ -4 & 4 & 1 & -1 \\ -4 & 4 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 4 & -4 \\ -1 & 1 & 4 & -4 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -4 & 4 \\ 1 & -1 & -4 & 4 \end{bmatrix}$$

$$[B3012] = \frac{1}{36}\begin{bmatrix} 4 & 2 & -2 & -4 \\ 2 & 4 & -4 & -2 \\ 1 & 2 & -2 & -1 \\ 2 & 1 & -1 & -2 \\ -4 & -2 & 2 & 4 \\ -2 & -4 & 4 & 2 \\ -1 & -2 & 2 & 1 \\ -2 & -1 & 1 & 2 \\ -2 & -1 & 1 & 2 \\ -1 & -2 & 2 & 1 \\ -2 & -4 & 4 & 2 \\ -4 & -2 & 2 & 4 \\ 2 & 1 & -1 & -2 \\ 1 & 2 & -2 & -1 \\ 2 & 4 & -4 & -2 \\ 4 & 2 & -2 & -4 \end{bmatrix}$$

$$[B3022] = \frac{1}{36}\begin{bmatrix} 4 & 1 & -1 & -4 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 4 & 1 & -1 & -4 \\ 1 & 1 & -1 & -1 \\ 1 & 4 & -4 & -1 \\ 1 & 4 & -4 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -4 & 4 & 1 \\ -1 & -4 & 4 & 1 \\ -1 & -1 & 1 & 1 \\ -4 & -1 & 1 & 4 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -4 & -1 & 1 & 4 \end{bmatrix}$$

$$[B3021] = \frac{1}{36}\begin{bmatrix} 4 & -4 & -2 & 2 \\ 2 & -2 & -1 & 1 \\ 1 & -1 & -2 & 2 \\ 2 & -2 & -4 & 4 \\ 2 & -2 & -1 & 1 \\ 4 & -4 & -2 & 2 \\ 2 & -2 & -4 & 4 \\ 1 & -1 & -2 & 2 \\ -2 & 2 & 1 & -1 \\ -4 & 4 & 2 & -2 \\ -2 & 2 & 4 & -4 \\ -1 & 1 & 2 & -2 \\ -4 & 4 & 2 & -2 \\ -2 & 2 & 1 & -1 \\ -1 & 1 & 2 & -2 \\ -2 & 2 & 4 & -4 \end{bmatrix}$$

$$[B40000] = \frac{1}{900}$$

$$\begin{bmatrix}
144 & 36 & 9 & 36 & 36 & 24 & 6 & 9 & 9 & 6 & 4 & 6 & 36 & 9 & 6 & 24 \\
36 & 24 & 6 & 9 & 24 & 36 & 9 & 6 & 6 & 9 & 6 & 4 & 9 & 6 & 4 & 6 \\
9 & 6 & 4 & 6 & 6 & 9 & 6 & 4 & 4 & 6 & 9 & 6 & 6 & 4 & 6 & 9 \\
36 & 9 & 6 & 24 & 9 & 6 & 4 & 6 & 6 & 4 & 6 & 9 & 24 & 6 & 9 & 36 \\
36 & 24 & 6 & 9 & 24 & 36 & 9 & 6 & 6 & 9 & 6 & 4 & 9 & 6 & 4 & 6 \\
24 & 36 & 9 & 6 & 36 & 144 & 36 & 9 & 9 & 36 & 24 & 6 & 6 & 9 & 6 & 4 \\
6 & 9 & 6 & 4 & 9 & 36 & 24 & 6 & 6 & 24 & 36 & 9 & 4 & 6 & 9 & 6 \\
9 & 6 & 4 & 6 & 6 & 9 & 6 & 4 & 4 & 6 & 9 & 6 & 6 & 4 & 6 & 9 \\
9 & 6 & 4 & 6 & 6 & 9 & 6 & 4 & 4 & 6 & 9 & 6 & 6 & 4 & 6 & 9 \\
6 & 9 & 6 & 4 & 9 & 36 & 24 & 6 & 6 & 24 & 36 & 9 & 4 & 6 & 9 & 6 \\
4 & 6 & 9 & 6 & 6 & 24 & 36 & 9 & 9 & 36 & 144 & 36 & 6 & 9 & 36 & 24 \\
6 & 4 & 6 & 9 & 4 & 6 & 9 & 6 & 6 & 9 & 36 & 24 & 9 & 6 & 24 & 36 \\
36 & 9 & 6 & 24 & 9 & 6 & 4 & 6 & 6 & 4 & 6 & 9 & 24 & 6 & 9 & 36 \\
9 & 6 & 4 & 6 & 6 & 9 & 6 & 4 & 4 & 6 & 9 & 6 & 6 & 4 & 6 & 9 \\
6 & 4 & 6 & 9 & 4 & 6 & 9 & 6 & 6 & 9 & 36 & 24 & 9 & 6 & 24 & 36 \\
24 & 6 & 9 & 36 & 6 & 4 & 6 & 9 & 9 & 6 & 24 & 36 & 36 & 9 & 36 & 144
\end{bmatrix}$$

$$[B40010] = \frac{1}{900}$$

$$\begin{bmatrix}
-36 & -12 & -3 & -9 & 36 & 12 & 3 & 9 & 9 & 3 & 2 & 6 & -9 & -3 & -2 & -6 \\
-12 & -12 & -3 & -3 & 12 & 12 & 3 & 3 & 3 & 3 & 2 & 2 & -3 & -3 & -2 & -2 \\
-3 & -3 & -2 & -2 & 3 & 3 & 2 & 2 & 2 & 2 & 3 & 3 & -2 & -2 & -3 & -3 \\
-9 & -3 & -2 & -6 & 9 & 3 & 2 & 6 & 6 & 2 & 3 & 9 & -6 & -2 & -3 & -9 \\
-12 & -12 & -3 & -3 & 12 & 12 & 3 & 3 & 3 & 3 & 2 & 2 & -3 & -3 & -2 & -2 \\
-12 & -36 & -9 & -3 & 12 & 36 & 9 & 3 & 3 & 9 & 6 & 2 & -3 & -9 & -6 & -2 \\
-3 & -9 & -6 & -2 & 3 & 9 & 6 & 2 & 2 & 6 & 9 & 3 & -2 & -6 & -9 & -3 \\
-3 & -3 & -2 & -2 & 3 & 3 & 2 & 2 & 2 & 2 & 3 & 3 & -2 & -2 & -3 & -3 \\
-3 & -3 & -2 & -2 & 3 & 3 & 2 & 2 & 2 & 2 & 3 & 3 & -2 & -2 & -3 & -3 \\
-3 & -9 & -6 & -2 & 3 & 9 & 6 & 2 & 2 & 6 & 9 & 3 & -2 & -6 & -9 & -3 \\
-2 & -6 & -9 & -3 & 2 & 6 & 9 & 3 & 3 & 9 & 36 & 12 & -3 & -9 & -36 & -12 \\
-2 & -2 & -3 & -3 & 2 & 2 & 3 & 3 & 3 & 3 & 12 & 12 & -3 & -3 & -12 & -12 \\
-9 & -3 & -2 & -6 & 9 & 3 & 2 & 6 & 6 & 2 & 3 & 9 & -6 & -2 & -3 & -9 \\
-3 & -3 & -2 & -2 & 3 & 3 & 2 & 2 & 2 & 2 & 3 & 3 & -2 & -2 & -3 & -3 \\
-2 & -2 & -3 & -3 & 2 & 2 & 3 & 3 & 3 & 3 & 12 & 12 & -3 & -3 & -12 & -12 \\
-6 & -2 & -3 & -9 & 6 & 2 & 3 & 9 & 9 & 3 & 12 & 36 & -9 & -3 & -12 & -36
\end{bmatrix}$$

$$[B40020] = \frac{1}{360}$$

$$\begin{bmatrix}
-36 & -9 & -3 & -12 & -9 & -6 & -2 & -3 & 9 & 6 & 2 & 3 & 36 & 9 & 3 & 12 \\
-9 & -6 & -2 & -3 & -6 & -9 & -3 & -2 & 6 & 9 & 3 & 2 & 9 & 6 & 2 & 3 \\
-3 & -2 & -2 & -3 & -2 & -3 & -3 & -2 & 2 & 3 & 3 & 2 & 3 & 2 & 2 & 3 \\
-12 & -3 & -3 & -12 & -3 & -2 & -2 & -3 & 3 & 2 & 2 & 3 & 12 & 3 & 3 & 12 \\
-9 & -6 & -2 & -3 & -6 & -9 & -3 & -2 & 6 & 9 & 3 & 2 & 9 & 6 & 2 & 3 \\
-6 & -9 & -3 & -2 & -9 & -36 & -12 & -3 & 9 & 36 & 12 & 3 & 6 & 9 & 3 & 2 \\
-2 & -3 & -3 & -2 & -3 & -12 & -12 & -3 & 3 & 12 & 12 & 3 & 2 & 3 & 3 & 2 \\
-3 & -2 & -2 & -3 & -2 & -3 & -3 & -2 & 2 & 3 & 3 & 2 & 3 & 2 & 2 & 3 \\
-3 & -2 & -2 & -3 & -2 & -3 & -3 & -2 & 2 & 3 & 3 & 2 & 3 & 2 & 2 & 3 \\
-2 & -3 & -3 & -2 & -3 & -12 & -12 & -3 & 3 & 12 & 12 & 3 & 2 & 3 & 3 & 2 \\
-2 & -3 & -9 & -6 & -3 & -12 & -36 & -9 & 3 & 12 & 36 & 9 & 2 & 3 & 9 & 6 \\
-3 & -2 & -6 & -9 & -2 & -3 & -9 & -6 & 2 & 3 & 9 & 6 & 3 & 2 & 6 & 9 \\
-12 & -3 & -3 & -12 & -3 & -2 & -2 & -3 & 3 & 2 & 2 & 3 & 12 & 3 & 3 & 12 \\
-3 & -2 & -2 & -3 & -2 & -3 & -3 & -2 & 2 & 3 & 3 & 2 & 3 & 2 & 2 & 3 \\
-3 & -2 & -6 & -9 & -2 & -3 & -9 & -6 & 2 & 3 & 9 & 6 & 3 & 2 & 6 & 9 \\
-12 & -3 & -9 & -36 & -3 & -2 & -6 & -9 & 3 & 2 & 6 & 9 & 12 & 3 & 9 & 36
\end{bmatrix}$$

$$[B40100] = \frac{1}{360}$$

$$\begin{bmatrix}
-36 & -12 & -3 & -9 & -12 & -12 & -3 & -3 & -3 & -3 & -2 & -2 & -9 & -3 & -2 & -6 \\
-12 & -12 & -3 & -3 & -12 & -36 & -9 & -3 & -3 & -9 & -6 & -2 & -3 & -3 & -2 & -2 \\
-3 & -3 & -2 & -2 & -3 & -9 & -6 & -2 & -2 & -6 & -9 & -3 & -2 & -2 & -3 & -3 \\
-9 & -3 & -2 & -6 & -3 & -3 & -2 & -2 & -2 & -2 & -3 & -3 & -6 & -2 & -3 & -9 \\
36 & 12 & 3 & 9 & 12 & 12 & 3 & 3 & 3 & 3 & 2 & 2 & 9 & 3 & 2 & 6 \\
12 & 12 & 3 & 3 & 12 & 36 & 9 & 3 & 3 & 9 & 6 & 2 & 3 & 3 & 2 & 2 \\
3 & 3 & 2 & 2 & 3 & 9 & 6 & 2 & 2 & 6 & 9 & 3 & 2 & 2 & 3 & 3 \\
9 & 3 & 2 & 6 & 3 & 3 & 2 & 2 & 2 & 2 & 3 & 3 & 6 & 2 & 3 & 9 \\
9 & 3 & 2 & 6 & 3 & 3 & 2 & 2 & 2 & 2 & 3 & 3 & 6 & 2 & 3 & 9 \\
3 & 3 & 2 & 2 & 3 & 9 & 6 & 2 & 2 & 6 & 9 & 3 & 2 & 2 & 3 & 3 \\
2 & 2 & 3 & 3 & 2 & 6 & 9 & 3 & 3 & 9 & 36 & 12 & 3 & 3 & 12 & 12 \\
6 & 2 & 3 & 9 & 2 & 2 & 3 & 3 & 3 & 3 & 12 & 12 & 9 & 3 & 12 & 36 \\
-9 & -3 & -2 & -6 & -3 & -3 & -2 & -2 & -2 & -2 & -3 & -3 & -6 & -2 & -3 & -9 \\
-3 & -3 & -2 & -2 & -3 & -9 & -6 & -2 & -2 & -6 & -9 & -3 & -2 & -2 & -3 & -3 \\
-2 & -2 & -3 & -3 & -2 & -6 & -9 & -3 & -3 & -9 & -36 & -12 & -3 & -3 & -12 & -12 \\
-6 & -2 & -3 & -9 & -2 & -2 & -3 & -3 & -3 & -3 & -12 & -12 & -9 & -3 & -12 & -36
\end{bmatrix}$$

$$[B40200] = \frac{1}{360}$$

$$\begin{bmatrix}
-36 & -9 & -3 & -12 & -9 & -6 & -2 & -3 & -3 & -2 & -2 & -3 & -12 & -3 & -3 & -12 \\
-9 & -6 & -2 & -3 & -6 & -9 & -3 & -2 & -2 & -3 & -3 & -2 & -3 & -2 & -2 & -3 \\
-3 & -2 & -2 & -3 & -2 & -3 & -3 & -2 & -2 & -3 & -9 & -6 & -3 & -2 & -6 & -9 \\
-12 & -3 & -3 & -12 & -3 & -2 & -2 & -3 & -3 & -2 & -6 & -9 & -12 & -3 & -9 & -36 \\
-9 & -6 & -2 & -3 & -6 & -9 & -3 & -2 & -2 & -3 & -3 & -2 & -3 & -2 & -2 & -3 \\
-6 & -9 & -3 & -2 & -9 & -36 & -12 & -3 & -3 & -12 & -12 & -3 & -2 & -3 & -3 & -2 \\
-2 & -3 & -3 & -2 & -3 & -12 & -12 & -3 & -3 & -12 & -36 & -9 & -2 & -3 & -9 & -6 \\
-3 & -2 & -2 & -3 & -2 & -3 & -3 & -2 & -2 & -3 & -9 & -6 & -3 & -2 & -6 & -9 \\
9 & 6 & 2 & 3 & 6 & 9 & 3 & 2 & 2 & 3 & 3 & 2 & 3 & 2 & 2 & 3 \\
6 & 9 & 3 & 2 & 9 & 36 & 12 & 3 & 3 & 12 & 12 & 3 & 2 & 3 & 3 & 2 \\
2 & 3 & 3 & 2 & 3 & 12 & 12 & 3 & 3 & 12 & 36 & 9 & 2 & 3 & 9 & 6 \\
3 & 2 & 2 & 3 & 2 & 3 & 3 & 2 & 2 & 3 & 9 & 6 & 3 & 2 & 6 & 9 \\
36 & 9 & 3 & 12 & 9 & 6 & 2 & 3 & 3 & 2 & 2 & 3 & 12 & 3 & 3 & 12 \\
9 & 6 & 2 & 3 & 6 & 9 & 3 & 2 & 2 & 3 & 3 & 2 & 3 & 2 & 2 & 3 \\
3 & 2 & 2 & 3 & 2 & 3 & 3 & 2 & 2 & 3 & 9 & 6 & 3 & 2 & 6 & 9 \\
12 & 3 & 3 & 12 & 3 & 2 & 2 & 3 & 3 & 2 & 6 & 9 & 12 & 3 & 9 & 36
\end{bmatrix}$$

$$[B40110] = \frac{1}{360}$$

$$\begin{bmatrix}
24 & 12 & 3 & 6 & -24 & -12 & -3 & -6 & -6 & -3 & -2 & -4 & 6 & 3 & 2 & 4 \\
12 & 24 & 6 & 3 & -12 & -24 & -6 & -3 & -3 & -6 & -4 & -2 & 3 & 6 & 4 & 2 \\
3 & 6 & 4 & 2 & -3 & -6 & -4 & -2 & -2 & -4 & -6 & -3 & 2 & 4 & 6 & 3 \\
6 & 3 & 2 & 4 & -6 & -3 & -2 & -4 & -4 & -2 & -3 & -6 & 4 & 2 & 3 & 6 \\
-24 & -12 & -3 & -6 & 24 & 12 & 3 & 6 & 6 & 3 & 2 & 4 & -6 & -3 & -2 & -4 \\
-12 & -24 & -6 & -3 & 12 & 24 & 6 & 3 & 3 & 6 & 4 & 2 & -3 & -6 & -4 & -2 \\
-3 & -6 & -4 & -2 & 3 & 6 & 4 & 2 & 2 & 4 & 6 & 3 & -2 & -4 & -6 & -3 \\
-6 & -3 & -2 & -4 & 6 & 3 & 2 & 4 & 4 & 2 & 3 & 6 & -4 & -2 & -3 & -6 \\
-6 & -3 & -2 & -4 & 6 & 3 & 2 & 4 & 4 & 2 & 3 & 6 & -4 & -2 & -3 & -6 \\
-3 & -6 & -4 & -2 & 3 & 6 & 4 & 2 & 2 & 4 & 6 & 3 & -2 & -4 & -6 & -3 \\
-2 & -4 & -6 & -3 & 2 & 4 & 6 & 3 & 3 & 6 & 24 & 12 & -3 & -6 & -24 & -12 \\
-4 & -2 & -3 & -6 & 4 & 2 & 3 & 6 & 6 & 3 & 12 & 24 & -6 & -3 & -12 & -24 \\
6 & 3 & 2 & 4 & -6 & -3 & -2 & -4 & -4 & -2 & -3 & -6 & 4 & 2 & 3 & 6 \\
3 & 6 & 4 & 2 & -3 & -6 & -4 & -2 & -2 & -4 & -6 & -3 & 2 & 4 & 6 & 3 \\
2 & 4 & 6 & 3 & -2 & -4 & -6 & -3 & -3 & -6 & -24 & -12 & 3 & 6 & 24 & 12 \\
4 & 2 & 3 & 6 & -4 & -2 & -3 & -6 & -6 & -3 & -12 & -24 & 6 & 3 & 12 & 24
\end{bmatrix}$$

$$[B40220] = \frac{1}{360}$$

$$\begin{bmatrix}
24 & 6 & 3 & 12 & 6 & 4 & 2 & 3 & -6 & -4 & -2 & -3 & -24 & -6 & -3 & -12 \\
6 & 4 & 2 & 3 & 4 & 6 & 3 & 2 & -4 & -6 & -3 & -2 & -6 & -4 & -2 & -3 \\
3 & 2 & 4 & 6 & 2 & 3 & 6 & 4 & -2 & -3 & -6 & -4 & -3 & -2 & -4 & -6 \\
12 & 3 & 6 & 24 & 3 & 2 & 4 & 6 & -3 & -2 & -4 & -6 & -12 & -3 & -6 & -24 \\
6 & 4 & 2 & 3 & 4 & 6 & 3 & 2 & -4 & -6 & -3 & -2 & -6 & -4 & -2 & -3 \\
4 & 6 & 3 & 2 & 6 & 24 & 12 & 3 & -6 & -24 & -12 & -3 & -4 & -6 & -3 & -2 \\
2 & 3 & 6 & 4 & 3 & 12 & 24 & 6 & -3 & -12 & -24 & -6 & -2 & -3 & -6 & -4 \\
3 & 2 & 4 & 6 & 2 & 3 & 6 & 4 & -2 & -3 & -6 & -4 & -3 & -2 & -4 & -6 \\
-6 & -4 & -2 & -3 & -4 & -6 & -3 & -2 & 4 & 6 & 3 & 2 & 6 & 4 & 2 & 3 \\
-4 & -6 & -3 & -2 & -6 & -24 & -12 & -3 & 6 & 24 & 12 & 3 & 4 & 6 & 3 & 2 \\
-2 & -3 & -6 & -4 & -3 & -12 & -24 & -6 & 3 & 12 & 24 & 6 & 2 & 3 & 6 & 4 \\
-3 & -2 & -4 & -6 & -2 & -3 & -6 & -4 & 2 & 3 & 6 & 4 & 3 & 2 & 4 & 6 \\
-24 & -6 & -3 & -12 & -6 & -4 & -2 & -3 & 6 & 4 & 2 & 3 & 24 & 6 & 3 & 12 \\
-6 & -4 & -2 & -3 & -4 & -6 & -3 & -2 & 4 & 6 & 3 & 2 & 6 & 4 & 2 & 3 \\
-3 & -2 & -4 & -6 & -2 & -3 & -6 & -4 & 2 & 3 & 6 & 4 & 3 & 2 & 4 & 6 \\
-12 & -3 & -6 & -24 & -3 & -2 & -4 & -6 & 3 & 2 & 4 & 6 & 12 & 3 & 6 & 24
\end{bmatrix}$$

$$[B40120] = \frac{1}{144}$$

$$\begin{bmatrix}
9 & 3 & 1 & 3 & 3 & 3 & 1 & 1 & -3 & -3 & -1 & -1 & -9 & -3 & -1 & -3 \\
3 & 3 & 1 & 1 & 3 & 9 & 3 & 1 & -3 & -9 & -3 & -1 & -3 & -3 & -1 & -1 \\
1 & 1 & 1 & 1 & 1 & 3 & 3 & 1 & -1 & -3 & -3 & -1 & -1 & -1 & -1 & -1 \\
3 & 1 & 1 & 3 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -3 & -1 & -1 & -3 \\
-9 & -3 & -1 & -3 & -3 & -3 & -1 & -1 & 3 & 3 & 1 & 1 & 9 & 3 & 1 & 3 \\
-3 & -3 & -1 & -1 & -3 & -9 & -3 & -1 & 3 & 9 & 3 & 1 & 3 & 3 & 1 & 1 \\
-1 & -1 & -1 & -1 & -1 & -3 & -3 & -1 & 1 & 3 & 3 & 1 & 1 & 1 & 1 & 1 \\
-3 & -1 & -1 & -3 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 3 & 1 & 1 & 3 \\
-3 & -1 & -1 & -3 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 3 & 1 & 1 & 3 \\
-1 & -1 & -1 & -1 & -1 & -3 & -3 & -1 & 1 & 3 & 3 & 1 & 1 & 1 & 1 & 1 \\
-1 & -1 & -3 & -3 & -1 & -3 & -9 & -3 & 1 & 3 & 9 & 3 & 1 & 1 & 3 & 3 \\
-3 & -1 & -3 & -9 & -1 & -1 & -3 & -3 & 1 & 1 & 3 & 3 & 3 & 1 & 3 & 9 \\
3 & 1 & 1 & 3 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -3 & -1 & -1 & -3 \\
1 & 1 & 1 & 1 & 1 & 3 & 3 & 1 & -1 & -3 & -3 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 3 & 3 & 1 & 3 & 9 & 3 & -1 & -3 & -9 & -3 & -1 & -1 & -3 & -3 \\
3 & 1 & 3 & 9 & 1 & 1 & 3 & 3 & -1 & -1 & -3 & -3 & -3 & -1 & -3 & -9
\end{bmatrix}$$

$$[B40210] = \frac{1}{144}$$

$$\begin{bmatrix}
9 & 3 & 1 & 3 & -9 & -3 & -1 & -3 & -3 & -1 & -1 & -3 & 3 & 1 & 1 & 3 \\
3 & 3 & 1 & 1 & -3 & -3 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -3 & -3 & 1 & 1 & 3 & 3 \\
3 & 1 & 1 & 3 & -3 & -1 & -1 & -3 & -3 & -1 & -3 & -9 & 3 & 1 & 3 & 9 \\
3 & 3 & 1 & 1 & -3 & -3 & -1 & -1 & -1 & -1 & -1 & -1 & 3 & 1 & 1 & 1 \\
3 & 9 & 3 & 1 & -3 & -9 & -3 & -1 & -1 & -3 & -3 & -1 & 1 & 3 & 3 & 1 \\
1 & 3 & 3 & 1 & -1 & -3 & -3 & -1 & -1 & -3 & -9 & -3 & 1 & 3 & 9 & 3 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -3 & -3 & 1 & 1 & 3 & 3 \\
-3 & -3 & -1 & -1 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
-3 & -9 & -3 & -1 & 3 & 9 & 3 & 1 & 1 & 3 & 3 & 1 & -1 & -3 & -3 & -1 \\
-1 & -3 & -3 & -1 & 1 & 3 & 3 & 1 & 1 & 3 & 9 & 3 & -1 & -3 & -9 & -3 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 3 & 3 & -1 & -1 & -3 & -3 \\
-9 & -3 & -1 & -3 & 9 & 3 & 1 & 3 & 3 & 1 & 1 & 3 & -3 & -1 & -1 & -3 \\
-3 & -3 & -1 & -1 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 3 & 3 & -1 & -1 & -3 & -3 \\
-3 & -1 & -1 & -3 & 3 & 1 & 1 & 3 & 3 & 1 & 3 & 9 & -3 & -1 & -3 & -9
\end{bmatrix}$$

APPENDIX C

FINITE ELEMENT ALGORITHM MATRICES
TRI-LINEAR TENSOR PRODUCT BASIS ON THREE-DIEMNSIONAL SPACE

$$[C200] = \frac{1}{8}\begin{bmatrix} 8 & 4 & 2 & 4 & 4 & 2 & 1 & 2 \\ 4 & 8 & 4 & 2 & 2 & 4 & 2 & 1 \\ 2 & 4 & 8 & 4 & 1 & 2 & 4 & 2 \\ 4 & 2 & 4 & 8 & 2 & 1 & 2 & 4 \\ 4 & 2 & 1 & 2 & 8 & 4 & 2 & 4 \\ 2 & 4 & 2 & 1 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 2 & 4 & 8 & 4 \\ 2 & 1 & 2 & 4 & 4 & 2 & 4 & 8 \end{bmatrix}$$

$$[C201] = \frac{1}{12}\begin{bmatrix} -4 & 4 & 2 & -2 & -2 & 2 & 1 & -1 \\ -4 & 4 & 2 & -2 & -2 & 2 & 1 & -1 \\ -2 & 2 & 4 & -4 & -1 & 1 & 2 & -2 \\ -2 & 2 & 4 & -4 & -1 & 1 & 2 & -2 \\ -2 & 2 & 1 & -1 & -4 & 4 & 2 & -2 \\ -2 & 2 & 1 & -1 & -4 & 4 & 2 & -2 \\ -1 & 1 & 2 & -2 & -2 & 2 & 4 & -4 \\ -1 & 1 & 2 & -2 & -2 & 2 & 4 & -4 \end{bmatrix}$$

$$[C202] = \frac{1}{12}\begin{bmatrix} -4 & -2 & 2 & 4 & -2 & -1 & 1 & 2 \\ -2 & -4 & 4 & 2 & -1 & -2 & 2 & 1 \\ -2 & -4 & 4 & 2 & -1 & -2 & 2 & 1 \\ -4 & -2 & 2 & 4 & -2 & -1 & 1 & 2 \\ -2 & -1 & 1 & 2 & -4 & -2 & 2 & 4 \\ -1 & -2 & 2 & 1 & -2 & -4 & 4 & 2 \\ -1 & -2 & 2 & 1 & -2 & -4 & 4 & 2 \\ -2 & -1 & 1 & 2 & -4 & -2 & 2 & 4 \end{bmatrix}$$

$$[C203] = \frac{1}{12}\begin{bmatrix} -4 & -2 & -1 & -2 & 4 & 2 & 1 & 2 \\ -2 & -4 & -2 & -1 & 2 & 4 & 2 & 1 \\ -1 & -2 & -4 & -2 & 1 & 2 & 4 & 2 \\ -2 & -1 & -2 & -4 & 2 & 1 & 2 & 4 \\ -4 & -2 & -1 & -2 & 4 & 2 & 1 & 2 \\ -2 & -4 & -2 & -1 & 2 & 4 & 2 & 1 \\ -1 & -2 & -4 & -2 & 1 & 2 & 4 & 2 \\ -2 & -1 & -2 & -4 & 2 & 1 & 2 & 4 \end{bmatrix}$$

$$[C210] = \frac{1}{12}\begin{bmatrix} -4 & -4 & -2 & -2 & -2 & -2 & -1 & -1 \\ 4 & 4 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 4 & 4 & 1 & 1 & 2 & 2 \\ -2 & -2 & -4 & -4 & -1 & -1 & -2 & -2 \\ -2 & -2 & -1 & -1 & -4 & -4 & -2 & -2 \\ 2 & 2 & 1 & 1 & 4 & 4 & 2 & 2 \\ 1 & 1 & 2 & 2 & 2 & 2 & 4 & 4 \\ -1 & -1 & -2 & -2 & -2 & -2 & -4 & -4 \end{bmatrix}$$

$$[C220] = \frac{1}{12}\begin{bmatrix} -4 & -2 & -2 & -4 & -2 & -1 & -1 & -2 \\ -2 & -4 & -4 & -2 & -1 & -2 & -2 & -1 \\ 2 & 4 & 4 & 2 & 1 & 2 & 2 & 1 \\ 4 & 2 & 2 & 4 & 2 & 1 & 1 & 2 \\ -2 & -1 & -1 & -2 & -4 & -2 & -2 & -4 \\ -1 & -2 & -2 & -1 & -2 & -4 & -4 & -2 \\ 1 & 2 & 2 & 1 & 2 & 4 & 4 & 2 \\ 2 & 1 & 1 & 2 & 4 & 2 & 2 & 4 \end{bmatrix}$$

$$[C230] = \frac{1}{12}\begin{bmatrix} -4 & -2 & -1 & -2 & -4 & -2 & -1 & -2 \\ -2 & -4 & -2 & -1 & -2 & -4 & -2 & -1 \\ -1 & -2 & -4 & -2 & -1 & -2 & -4 & -2 \\ -2 & -1 & -2 & -4 & -2 & -1 & -2 & -4 \\ 4 & 2 & 1 & 2 & 4 & 2 & 1 & 2 \\ 2 & 4 & 2 & 1 & 2 & 4 & 2 & 1 \\ 1 & 2 & 4 & 2 & 1 & 2 & 4 & 2 \\ 2 & 1 & 2 & 4 & 2 & 1 & 2 & 4 \end{bmatrix}$$

$$[C211] = \frac{1}{8}\begin{bmatrix} 4 & -4 & -2 & 2 & 2 & -2 & -1 & 1 \\ -4 & 4 & 2 & -2 & -2 & 2 & 1 & -1 \\ -2 & 2 & 4 & -4 & -1 & 1 & 2 & -2 \\ 2 & -2 & -4 & 4 & 1 & -1 & -2 & 2 \\ 2 & -2 & -1 & 1 & 4 & -4 & -2 & 2 \\ -2 & 2 & 1 & -1 & -4 & 4 & 2 & -2 \\ -1 & 1 & 2 & -2 & -2 & 2 & 4 & -4 \\ 1 & -1 & -2 & 2 & 2 & -2 & -4 & 4 \end{bmatrix}$$

$$[C222] = \frac{1}{8}\begin{bmatrix} 4 & 2 & -2 & -4 & 2 & 1 & -1 & -2 \\ 2 & 4 & -4 & -2 & 1 & 2 & -2 & -1 \\ -2 & -4 & 4 & 2 & -1 & -2 & 2 & 1 \\ -4 & -2 & 2 & 4 & -2 & -1 & 1 & 2 \\ 2 & 1 & -1 & -2 & 4 & 2 & -2 & -4 \\ 1 & 2 & -2 & -1 & 2 & 4 & -4 & -2 \\ -1 & -2 & 2 & 1 & -2 & -4 & 4 & 2 \\ -2 & -1 & 1 & 2 & -4 & -2 & 2 & 4 \end{bmatrix}$$

$$[C233] = \frac{1}{8}\begin{bmatrix} 4 & 2 & 1 & 2 & -4 & -2 & -1 & -2 \\ 2 & 4 & 2 & 1 & -2 & -4 & -2 & -1 \\ 1 & 2 & 4 & 2 & -1 & -2 & -4 & -2 \\ 2 & 1 & 2 & 4 & -2 & -1 & -2 & -4 \\ -4 & -2 & -1 & -2 & 4 & 2 & 1 & 2 \\ -2 & -4 & -2 & -1 & 2 & 4 & 2 & 1 \\ -1 & -2 & -4 & -2 & 1 & 2 & 4 & 2 \\ -2 & -1 & -2 & -4 & 2 & 1 & 2 & 4 \end{bmatrix}$$